

Tan, Kay Chen (1997) *Evolutionary methods for modelling and control of linear and nonlinear systems*. PhD thesis.

<http://theses.gla.ac.uk/8473/>

Copyright and moral rights for this thesis are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given



**UNIVERSITY  
of  
GLASGOW**

**EVOLUTIONARY METHODS FOR MODELLING AND  
CONTROL OF LINEAR AND NONLINEAR SYSTEMS**

by

**Kay Chen Tan**

Thesis submitted to the

**Faculty of Engineering**

**UNIVERSITY OF GLASGOW**

for the degree of

**DOCTOR OF PHILOSOPHY**

**© KAY CHEN TAN (APRIL 1997)**



謹獻給：

我的父母，姐姐，兄弟和俐萍

## Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Yun Li for his guidance, encouragement, inspiration and patience throughout the course of my Ph.D. work. In particular, his vision in, and commitments to, the research and his sharing of knowledge and time that has made the research a success.

Special acknowledgement and appreciation are due to Mr. Tom O'Hara for his enthusiastic assistance during my laboratory work.

I would like to thank my family in Malaysia, especially my father Cher Kee Tan, my mother Kim Hoi, my brothers, sister and my girlfriend Lay Peng. Without their love and moral support, nothing would have been achieved.

I am also grateful to the University of Glasgow and CVCP for their financial support in the form of a Postgraduate Scholarship and an Overseas Research Scheme Award, respectively.

Finally, I want to thank all the friends in Evolutionary Computing and Control group for their friendliness, help and companionship. In particular, Dr. Kim Chwee Ng, Dr. Ming Rui Gong, Dr. Gary Gray, Henrik Gollee, Dr. Graham Dudgeon, Jay Sigbrandt, Munir Chowdhury and Wen Yuan.



## ABSTRACT

The aim of this work is to explore the potential and enhance the capability of evolutionary computation for the development of novel and advanced methodologies for engineering system modelling and controller design automation. The key to these modelling and design problems is optimisation.

Conventional calculus-based methods currently adopted in engineering optimisation are in essence local search techniques, which require derivative information and lack of robustness in solving practical engineering problems. One objective of this research is thus to develop an effective and reliable evolutionary algorithm for engineering applications. For this, a hybrid evolutionary algorithm is developed, which combines the global search power of a “generational” EA with the interactive local fine-tuning of Boltzmann learning. It overcomes the weakness in local exploration and chromosome stagnation usually encountered in pure EAs. A novel one-integer-one-parameter coding scheme is also developed to significantly reduce the quantisation error, chromosome length and processing overhead time. An “Elitist Direct Inheritance” technique is developed to incorporate with Boltzmann learning for reducing the control parameters and convergence time of EAs. Parallelism of the hybrid EA is also realised in this thesis with nearly linear pipelinability.

Generic model reduction and linearisation techniques in  $L_2$  and  $L_\infty$  norms are developed based on the hybrid EA technique. They are applicable to both discrete and continuous-time systems in both the time and the frequency domains. Superior to conventional model reduction methods, the EA based techniques are capable of simultaneously recommending both an optimal order number and optimal parameters by a control gene used as a structural switch. This approach is extended to MIMO system linearisation from both a nonlinear model and I/O data of the plant. It also allows linearisation for an entire operating region with the linear approximate-model network technique studied in this thesis.

To build an original model, evolutionary black-box and clear-box system identification techniques are developed based on the  $L_2$  norm. These techniques can identify both the system parameters and transport delay in the same evolution process. These open-loop identification methods are further extended to closed-loop system identification. For robust control, evolutionary  $L_\infty$  identification techniques are developed. Since most practical systems are nonlinear in nature and it is difficult to model the dominant dynamics of such a system while retaining neglected dynamics for accuracy, evolutionary grey-box modelling techniques are proposed. These techniques can utilise physical law dominated global clear-box structure, with local black-boxes to include unmeasurable nonlinearities as the coefficient models of the clear-box. This unveils a new way of engineering system modelling.

With an accurately identified model, controller design problems still need to be overcome. Design difficulties by conventional analytical and numerical means are discussed and a design automation technique is then developed. This is again enabled by the hybrid evolutionary algorithm in this thesis. More importantly, this technique enables the unification of linear control system designs in both the time and the frequency domains under performance satisfaction. It is also extended to control along a trajectory of operating points for nonlinear systems. In addition, a multi-objective evolutionary algorithm is developed to make the design more transparent and visible. To achieve a step towards autonomy in building control systems, a technique for direct designs from plant step response data is developed, which bypasses the system identification phase. These computer-automated intelligent design methodologies are expected to offer added productivity and quality of control systems.

## LIST OF PUBLICATIONS

- [1] Li, Y., Tan, K.C. and Gong, M.R., (1997). Global structure evolution and local parameter learning for control system model reductions, *Evolutionary Algorithms in Engineering Applications*, D. Dasgupta and Z. Michalewicz (Eds.), Springer Verlag, 345-360.
- [2] Tan, K.C., Li, Y., Gawthrop, P.J. and Glidle, A., (1997). Evolutionary grey-box modelling for practical systems, *Second IEE Int. Conf. on GA in Eng. Syst. Innovations and Appl.*, Glasgow, UK. (accepted)
- [3] Tan, K.C. and Li, Y., (1997). Multi-objective genetic algorithm based time and frequency domain design unification of control systems, *IFAC Int. Sym. on Artificial Intelligence in Real-Time Control*, Kuala Lumpur, Malaysia. (accepted)
- [4] Tan, K.C., Gong, M.R. and Li, Y., (1996). Evolutionary linearisation in the frequency domain, *Electronics letters.*, **32**, (1), 74-76.
- [5] Tan, K.C. and Li, Y., (1996).  $L_2$  identification and model reduction using a learning genetic algorithm, *Int. Conf. on Control'96*, University of Exeter, UK, 1125-1130.
- [6] Li, Y., Tan, K.C. and Marionneau, C. (1996). Direct design of linear control systems from plant I/O data using parallel evolutionary algorithms, *Int. Conf. on Control'96, Special Session on Evolutionary Algorithms for Control Engineering*, University of Exeter, UK, 680-686.
- [7] Tan, K.C., Li, Y., Murray-Smith, D.J. and Sharman, K.C., (1995). System identification and linearisation using genetic algorithms with simulated annealing. *First IEE/IEEE Int. Conf. on GA in Eng. Syst. Innovations and Appl.*, Sheffield, UK, 164-169.
- [8] Li, Y., Tan, K.C., Ng, K.C. and Murray-Smith, D.J., (1995). Performance based linear control system design by genetic evolution with simulated annealing, *34th IEEE Conf. Decision and Control*, New Orleans, LA, 731-736.
- [9] Li, Y., Ng, K.C., Tan, K.C., Gray, G.J., McGookin, E.W., Murray-Smith, D.J. and Sharman, K.C., (1995). Automation of linear and nonlinear control systems design by evolutionary computation, *Proc. IFAC Youth Automation Conf.*, Beijing, China, 53-58.

# CONTENTS

<b>Abstract</b>	iv
<b>List of Publications</b>	vi
<b>Chapter 1</b>	
<b>Introduction</b>	1
1.1 Engineering System Modelling and Existing Difficulties	1
1.2 Control System CAD and Design Automation Difficulties	3
1.3 Evolutionary Methodology for Modelling and Design Automation	4
1.4 Organisation of the Thesis	7
<b>Chapter 2</b>	
<b>Evolutionary Algorithms and Enhancement by Learning</b>	10
2.1 Evolutionary Search Algorithms	10
2.2 Non-Binary Coding Schemes for System Parameters and Structures	13
2.3 Genetic Operators	15
2.3.1 Selection and Reproduction Schemes	15
2.3.1.1 Roulette Wheel Selection	15
2.3.1.2 Rank-Based Selection	16
2.3.1.3 Tournament Selection	17
2.3.2 Crossover	18
2.3.3 Mutation	19
2.4 Evolutionary Building Blocks and the Schema Theorem	19
2.5 Other Techniques in EA Operations	20
2.5.1 Niche and Speciation	20
2.5.2 Mating Restriction	23
2.5.3 Generation Gap	23
2.5.4 Parallelism	24
2.6 Multi-Objective Optimisation	25
2.7 Improving EA Operations	29
2.7.1 Existing EA Problems	29
2.7.2 Fitness Functions for Control Engineering Problems	30
2.7.3 One-Integer-One-Parameter Coding Scheme	31
2.7.4 Direct Inheritance as Selection and Crossover	32
2.7.5 Transputer Based Parallel EA	33
2.8 Boltzmann Learning to Enhance Heterogeneous and Local Search	35
2.8.1 The Algorithm	35
2.8.2 Benchmark Test	40
2.9 Multi-Objective Evolutionary Algorithms	44

2.9.1 A Tournament Niche MOEA	44
2.9.2 Optimisation by Tournament Niche MOEA	45
2.10 Summary	48

## Chapter 3

<b>Highly Accurate Model Reduction and Linearisation by Learning and Evolution</b>	<b>51</b>
3.1 The Model Order Reduction and Linearisation Problem	51
3.2 $L_2$ Model Reduction by Evolution	53
3.2.1 Reduction Based on a Time Domain Cost	53
3.2.2 A Time Domain Model Reduction Example	55
3.2.3 Reduction Based on a Frequency Domain Cost	58
3.2.4 Open-Order Model Reduction in the Frequency Domain	59
3.3 $L_2$ Linearisation: Model Reduction from an Infinite-Order	63
3.3.1 Time Domain $L_2$ Linearisation	63
3.3.2 Frequency Domain $L_2$ Linearisation	75
3.4 $L_\infty$ Model Reduction by Evolution	79
3.5 $L_\infty$ Linearisation: Model Reduction from an Infinite-Order	82
3.6 Summary	83

## Chapter 4

<b>Evolutionary System Identification and Modelling</b>	<b>85</b>
4.1 The System Identification and Modelling Problem	85
4.2 Efficient and Accurate Evolution Based Linear System Identification	87
4.2.1 Evolutionary $L_2$ System Identification	87
4.2.1.1 ARMAX Based Black-Box Model	87
4.2.1.2 ODE Based Clear-Box Model	92
4.2.2 Evolutionary $L_\infty$ System Identification	94
4.3 Efficient and Accurate Evolution Based Nonlinear Clear-Box Identification	100
4.3.1 A Coupled Nonlinear System and Measured Clear-Box Model	100
4.3.2 Optimisation of the Clear-Box Model	102
4.4 Evolutionary Grey-Box Modelling for Practical Systems	104
4.4.1 Grey-Box Model of the Coupled Nonlinear System	105
4.4.2 A Measurement System Modelling Problem	109
4.5 Evolution Based Closed-Loop System Identification	114
4.5.1 Closed-Loop Identification for a Linear System	115
4.5.2 Closed-Loop Identification for a Nonlinear System	117
4.6 Summary	118

**Chapter 5**

<b>Unification and Automation of Linear Controller Design for Linear and Nonlinear Plants</b>	<b>121</b>
5.1 The Problem of Design	121
5.1.1 Design Specification and Problem Formation	122
5.1.2 Problem Classification	124
5.1.3 Unsolvability of Practical Design Problem by Conventional Means	125
5.1.4 Solvable Problem by Non-NP Numerical Means	127
5.2 Design Automation by Learning and Evolution	128
5.2.1 The Methodology	128
5.2.2 Limitations of the Evolutionary Methodology	130
5.3 Unification of LTI Control System Designs	130
5.3.1 Performance Based Design	130
5.3.2 Building Blocks of Cost Functions for ULTIC	133
5.4 Design Automation of ULTIC Systems for Linear Plants	138
5.4.1 ULTIC for a Delayed Linear Plant	138
5.4.1.1 Performance Based Time-Domain Design	139
5.4.1.2 Design with Emphasis on Robust Performance	142
5.4.2 ULTIC for a Nonlinear Plant Bypassing Linearisation	145
5.4.2.1 Design for a Single Operating Point	145
5.4.2.2 Design for an Operating Trajectory	148
5.4.3 Model Following Control	150
5.5 Multiple Input and Multiple Output ULTIC	152
5.6 Multi-Objective ULTIC Systems	155
5.7 Parallel EA and Near-Linear Pipelinability	159
5.8 Summary	160

**Chapter 6**

<b>Towards Autonomy: Evolving Controllers Directly from Open-Loop Response Data</b>	<b>162</b>
6.1 Step Response Data Represent a High-Fidelity Model	162
6.2 Evaluating ULTIC System Performance from Plant Step Response Data	164
6.3 Evolving an ULTIC System for a Linear Plant Directly from Open-Loop Response Data	164
6.4 Evolving an ULTIC System for an Unseen Nonlinear Plant Directly from Open-Loop Response Data	167
6.5 Summary	168

**Chapter 7**

<b>Conclusion and Further Work</b>	169
7.1 Application and Enhancement of Evolutionary Algorithms	169
7.2 EA Based Methodologies for Model Reduction and Linearisation	170
7.3 EA Based Methodologies for System Identification and Modelling	171
7.4 EA Based Methodology for LTI Control System Design Unification and Automation	171
7.5 Future Perspectives	172
<b>References</b>	174
<b>Appendix A</b>	183
<b>Appendix B</b>	186

## Chapter 1

# INTRODUCTION

### 1.1. Engineering System Modelling and Existing Difficulties

With rapid developments in classical and modern control theory and technology in the past few decades, a control engineer has now had better opportunities to devise and exploit better and advanced control for more and demanding applications. In doing so, the first challenge usually facing the engineer is to obtain an accurate model of the plant to be controlled, which can be complex and multivariable in nature.

For such a system identification and modelling task, a multi-frequency or pseudo random binary sequence (PRBS) signal is applied to the plant and the plant output is collected. The plant input-output (I/O) data are then used to identify the system in an optimisation process. For this, conventional optimisation methods based upon least mean-squares (LMS) or maximum likelihood estimates have been developed in the past few decades to build a mathematical model of the dynamic system.

System identification was first applied to the determination of a “black-box” transfer function of linear plants. For example, a linear black-box ARMAX model was first identified by Åström and Bohlin (1965) using the maximum likelihood estimation method and thereafter a wide range of  $L_2$  norm based identification and modelling techniques were developed for both continuous and discrete-time systems. These system identification techniques have flourished in both the time and the frequency domains (Beck and Arnold, 1977; Gawthrop *et al.*, 1989; Ljung, 1987; Söderström and Stoica, 1989). With the success in self-tuning and modern robust control, extending these techniques to on-line recursive and worst-case (or  $L_\infty$  norm) system identification for linear and nonlinear systems have also been widely developed (Åström and Wittenmark, 1989; Davies, 1970; Gawthrop *et al.*, 1989; Helmicki *et al.*, 1991; Ljung and Söderström, 1983).



The area of system identification and modelling has been fairly mature, but many problems still exist. The conventional, calculus-based numerical methods used in system identification are in essence gradient-guided search techniques. These techniques require a smooth search space or a differentiable cost function and are local, as opposed to global, optimisation techniques (Goldberg, 1989). They work well only if (1) the gradients of the cost function are well defined; (2) the search space is uni-modal; and (3) the dimension of the search space is small (Flexible Intelligence Group, 1995). However, in practice, the search index is not always “well-behaved” or differentiable. Thus, a calculus-based approach can easily fail in obtaining the global optimum and lack robustness in solving usually noisy, and multi-dimensional, engineering optimisation problems.

In summary, conventional identification methods can suffer from the following drawbacks:

- The system structure usually needs to be known *a-priori*, information on the initial values or starting points of some parameters is also required (Kristinsson and Dumont, 1992);
- The estimated parameters may be biased if measurement or process noise is correlated and inappropriate cost function is adopted (Sharman, *et al.*, 1995);
- It is difficult to identify the transport delay (Ljung, 1992);
- They cannot be applied easily to nonlinear systems especially when the parameters are not linearly separable (Vandemolengraft *et al.*, 1994);
- Identification process may fail if the identification matrix contains many zero elements, i.e., is sparse or singular (Tan *et al.*, 1995).

Thus, the above problems existing in engineering system identification and modelling need to be solved, so that the quality and productivity of dynamic models and of the controllers to be built upon them can be improved.

## 1.2 Control System CAD and Design Automation Difficulties

Once the model of a plant is obtained, design of its control system can begin. Most control systems used in industry employ linear schemes, which range from the most straightforward proportional plus integral plus derivative (PID), phase lead/lag and pole-placement control schemes to more sophisticated optimal, adaptive and robust schemes. Such a linear control system is usually designed in the form of a transfer function or transfer function matrix in terms of the Laplace transform (for continuous-time systems) or z-transform (for discrete-time systems) or in the form of a set of state-space ordinary differential equations (ODEs). In control engineering practice using classical or modern control theory, both the selection of an appropriate control scheme for the application and the selection of the best parameter set for the controller need to be made before any practical implementations are attempted.

Similar to system identification and modelling, a control system design problem is usually a multi-objective optimisation problem in a multi-dimensional space. Although modern control schemes such as  $H_\infty$  control can yield an optimal or sub-optimal controller by solving two Riccati equations, the resulting controller is always of very high order and is difficult for on-line implementation in practice (Chiang and Safonov, 1992; Doyle *et al.*, 1992). In addition, such a frequency domain design method is confined only to a narrow problem domain and it is difficult to take into account time domain constraints of practical systems, such as voltage/current limits. Thus, such a controller may lead to system degradation or may not realise the full potential of the controller when implemented.

Complexity, nonlinearity and constraints in practical systems, such as voltage/current limits, saturation, backlash, transport delays and noise make the design problem difficult to solve using calculus-based analytical or conventional numerical techniques. This has contributed to the reason that many Computer-Aided Control System Design (CACSD) packages currently available for industrial applications are mainly simulation tools which accommodate few direct or automated design facilities (Li *et al.*, 1995a). Using such a CACSD package in design, if the simulated performance is not satisfactory, the designer needs to adjust the parameters manually and carry out the simulation process again, until a "satisfactory" design emerges. Clearly, such a design process cannot be carried out easily,

since mutual interactions among multiple parameters are hard to predict. Also, a different control scheme requires a different design philosophy. All these make it difficult to achieve a computer-automated design that provides highest possible performance and best meets the design specifications automatically.

In summary, the limitations and drawbacks of existing methods and technology are (Li, *et al.*, 1995a):

- Manual adjustment using CACSD is difficult and the design process is usually tediously long;
- The designed controller is restricted to the control scheme that is selected *a-priori*, which may not best suit the application for required specifications; and
- The performance of the final controller may be far from the best and room is left for further adjustment to realise the full potential.

The manual trial-and-error technique does, however, indicate that an unsolvable design problem can be transformed into an exhaustive evaluation problem that is solvable in 'exponential time' by a deterministic search algorithm (Li, 1996). Although the transformed problem would not be practically solved for multi-parameter systems, it may be further transformed into a tractable 'NP-complete' problem by a 'non-deterministic polynomial' (NP) algorithm (Li, 1996; Michalewicz, 1994). A 'NP-complete' problem is defined as problems that cannot be solved by any deterministic algorithms in polynomial time but can be solved by a non-deterministic algorithm in polynomial time (Sedgewick, 1988). An evolutionary algorithm (EA) is just such an NP algorithm that can solve the optimisation problem in 'polynomial time' and should thus enable practical design automation of control systems (Li, 1996). The EA techniques form the underlying methodology of the work reported in this thesis.

### **1.3 Evolutionary Methodology for Modelling and Design Automation**

Based on Charles Darwin's biological observations, the means of natural selection and the principle of *survival-of-the-fittest* have led to today's success in evolutionary computation

(Fogel, 1995; Goldberg, 1989; Holland, 1975; Michalewicz, 1994). Evolutionary algorithms (EA) such as Genetic Algorithms (GAs), Genetic Programming (GP), Evolutionary Programming (EP), Evolutionary Strategies (ES) and related artificial life strategies have been developed upon the synthesis of natural evolution. They form the paradigm of evolutionary computation and have been found particularly effective in searching poorly understood, irregular and complex spaces for optimisation and machine learning. Unlike conventional gradient-guided search techniques, which are *a-priori*, EAs are *a-posteriori* and require no derivative information of the search points. These algorithms are probabilistic in nature and, based on *a-posteriori* information obtained by computerised trial-and-error, require no direct guidance and thus no stringent conditions on the cost function (Fogel, 1995; Michalewicz 1994). Therefore, the index function can be constructed in a way that satisfies the need of engineering systems most and not the need of analytical or numerical tools to be employed.

EAs exhibit global search capabilities by simultaneously evaluating performances at multiple points in the solution space. Supported by the *Schema Theory* (Goldberg, 1989; Holland, 1975), it has been shown that evolutionary algorithms offer an exponentially reduced search time in the order of  $O(n^m)$ ,  $m < \infty$ , compared with exhaustive search, which requires a total evaluation time of  $O(p^n)$ , where  $n$  being the number of parameters to be optimised in the search and  $p$  the number of possible choices of each parameter. EAs can handle multiple objectives (MO) without the need to define a composite scalar objective function (Goldberg, 1989). In addition, different preference of the multi-objective components is also allowed to make the optimisation transparent (Fonseca and Fleming, 1993) and suitable for 'minimum-commitment' at the CAD stage (Tan and Li, 1997b). The multiple search nature of reproductive and evolving population indicates that EAs are a natural parallel paradigm (Goldberg, 1989; Li *et al.*, 1997). Other features of EAs include robustness of search, capability to incorporate *a-priori* knowledge and adaptability (Goldberg, 1989; Li, 1996; Michalewicz, 1994). These NP algorithms could become even more reliable and accurate if interactive fine-tuning such as Boltzmann learning is incorporated (Tan *et al.*, 1995). The evolution process can also be speeded up several times when existing design experience is included in the initial design 'database' for intelligent design-reuse (Ng, 1995).

To characterise, EAs differ from conventional optimisation and search algorithms in several ways (Goldberg, 1989):

- EAs use probabilistic rules to make decisions. This has introduced intellectual capability in EAs and transforms a deterministic problem to a non-deterministic.
- EAs evaluate multiple points in the solution space simultaneously, instead of a single point. Therefore, it is capable of avoiding many local optima.
- EAs use payoff (objective function) information to guide the search and thus they are more robust in achieving optimal solution compare to *a-priori* optimisation techniques.

In summary, EAs have been found to be very effective and powerful in searching poorly understood, irregular, complex or non-differentiable spaces for optimisation and machine learning (Goldberg, 1989; Holland, 1975). They can provide feasible automated solutions to system modelling and design with the capability to meet multi-objective criteria simultaneously and to deal with complex problems robustly. Control engineers' existing knowledge and experience can be included in the EA to assist a faster modelling or design. The use of coding in an EA enables possible representation and adjustment of a system structure and parameters of the structure, in the same evolution process. This technique has been successfully applied to the general area of information technology and, in particular, control systems modelling (Li, 1995; Sharman and Esparcia-Alcázar, 1993; Tan *et al.*, 1997), identification (Fonseca *et al.*, 1993; Kristinsson and Dumont, 1992; Tan *et al.*, 1995; Yang *et al.*, 1997; Yao and Sethares, 1994), model order reduction (Li *et al.*, 1997; Tan and Li, 1996), linearisation (Tan *et al.*, 1995, 1996), controller order reduction (Caponetto *et al.*, 1994), auto-tuning PID (Wang and Kwok, 1992), optimal control (Fleming and Fonseca, 1993; Hunt 1992), linear control system unification and design automation (Li *et al.*, 1995a, 1995b; Li *et al.*, 1996b; Tan and Li, 1997b), robust control and stability analysis (Dakev *et al.*, 1995; Goh *et al.*, 1996; Hunt, 1992; Murdock *et al.*, 1991; Patton and Liu, 1994), fault detection (Patton *et al.*, 1995), fuzzy logic control (Karr, 1992; Linkens and Abbod, 1992; Ng, 1995), neural network control (Li and Häußler, 1996; Ng, 1995) and sliding mode control (Li *et al.*, 1996a; Ng, 1995).

## 1.4 Organisation of the Thesis

Since designs of many control systems require accurate and robust plant identification and modelling, most issues discussed in the previous sections are to be addressed in this thesis. The work seeks to explore the potential of artificial evolution based optimisation techniques for developing novel and advanced methodologies for control systems engineering. They range from model reduction, linearisation, identification and modelling to linear control system design and unification, design for nonlinear plants and design directly from plant response data.

In this work, one objective is to develop an effective and reliable evolutionary algorithm. Chapter 2 presents the achievement of a hybrid EA that combines the global search power of a “generational” EA with the interactive local fine-tuning of Boltzmann learning. Necessary background of EAs are discussed first, followed by more advanced techniques such as niching, mating restriction, parallelism and multi-objective optimisation. Various enhancement features to improve performance of the EAs are then presented. A one-integer-one-parameter coding scheme is proposed to reduce the chromosome length and to improve the computation efficiency of EAs. An “Elitist Direct Inheritance” technique that goes beyond conventional reproduction and crossover operations is proposed to reduce the EA *control parameters* and to improve the convergence of the EAs. A Multi-objective evolutionary algorithm (MOEA) that implements tournament selection based on Pareto’s optimality cost-assignment is also studied in Chapter 2. Niching and mating restriction techniques are applied to the MOEA for a better evolution along the Pareto optimal surface. Owing to the EA feature of multiple search points and multiple candidate solutions, an island model parallel EA implemented on a network of transputers of a Parsytec SuperCluster will also be discussed.

Chapter 3 develops evolutionary  $L_2$  and  $L_\infty$  norm based model reduction techniques for control engineering applications. The approach is targeted to generic applications to both discrete and continuous-time single-input single-output (SISO) and multiple-input multiple-output (MIMO) systems. Enabled by a control gene for structural switch, it is also for simultaneously recommending both an optimal order number and optimal parameters. It is then extended to MIMO nonlinear system linearisation in both the time and the

frequency domains. This technique is further extended to straightforward linear model approximation, inverse model linearisation and linearisation by linear approximate-model network which allows linearisation for an entire operating region.

Chapter 4 studies an  $L_2$  norm based evolutionary black-box identification method for a benchmark discrete-time dynamic system. The benchmark system is described by an ARMAX model in the presence of white noise. Evolutionary ODE based clear-box identification for a linear DC servo-system and a nonlinear coupled liquid-level regulation system are also studied. Evolutionary  $L_\infty$  norm based black-box identification for robust control applications is also studied in Chapter 4. The technique is to obtain an optimal nominal model and an uncertainty bounding function for both continuous and discrete-time systems. To overcome the deficiency of operating point ‘dependency’ occurred in a clear-box model, a grey-box modelling technique is developed, which makes the best use of *a-priori* knowledge on the global clear-box structure of a physical system whilst incorporating accurate black-boxes for unmeasurable local nonlinearities. Evolution based closed-loop system identification is also studied in this Chapter. The technique is to allow refinements of a clear-box model identified from plant response data, within a relatively short processing time, such that on-line tuning of the nonlinear model may be made.

Having obtained a suitable model, automation of linear control system designs by evolutionary computation is demonstrated in Chapter 5. Such design automation is to achieve unification of linear control system designs by unifying LTI approaches in both the time and frequency domains under performance satisfactions. The performance based Uniform Linear Time Invariant Control (ULTIC) technique also aims at easy applications without manual calculations or *a-priori* knowledge, while accommodating many requirements imposed by practical specifications for both SISO and MIMO systems. This technique is extended to control along a trajectory of operating points of a nonlinear system. Apart from following a fixed step reference, time-varying reference based design is also realised for the ULTIC system. A multi-objective evolutionary algorithm is incorporated in the ULTIC design in Chapter 5, to enable control engineers to interplay with design objectives and satisfying the ‘minimum-commitment’ principle at the CAD stage. In this Chapter, a Parallel EA is also implemented to assess the effectiveness of the parallelism in the ULTIC design.

Chapter 6 addressed the design automation issues for ULTIC systems by efficient evolution from plant step response data, bypassing the system identification stage. The underlying aim is to let a control engineer conveniently obtain an “off-the-computer” controller directly from the plant step response data and his/her building blocks of customer specifications. This offers a step towards autonomy in generating digital control systems.

This research has aimed to develop an intelligent methodology that is applicable to a generic class of engineering system modelling and control problems. Thus ad hoc case studies using pure mathematical representation of an industrial plant are not the focus of the thesis. However, the author views that validations using physical measurements are important to the methodological development. Hence all methods proposed are to be tested, via physical implementation. Results and conclusions are summarised in Chapter 7, together with suggested future work.



## Chapter 2

# EVOLUTIONARY ALGORITHMS AND ENHANCEMENT BY LEARNING

### 2.1 Evolutionary Search Algorithms

The most widely applied evolutionary algorithm (EA) is the genetic algorithm (GA), a coding version of EAs. For a control system problem, the “genetic codes” enable possible representation and adjustment of the system structure in addition to parameters of the structure in the same evolution process (Li *et al.*, 1997). This algorithm employs all three “genetic operators”, namely, *selection*, *crossover* and *mutation*. Although another operator “inversion” is also useful, it is used much less since inversion may be derived from crossover and mutation (Michalewicz, 1994). A GA generally uses coded strings (*chromosomes*) of binary numbers (*genes*) for the search process. Such an algorithm is based on an analogy with the genetic code in our own DNA structure, where the coded chromosome is composed of many genes having 64 values ( $64 = 4^3$  being the total number of different *words* permuted from 3 different alphabets out of A, C, G and T representing the 4 nitrogen-containing *bases*). For engineering applications, coding using a finite string-length in a GA will limit resolutions or ranges of the real-valued system parameters, but the use of an additional “control gene” in coding can provide a dimension for switching between possible parameter ranges and system structures (Li *et al.*, 1997).

The initial population of parameter sets can be generated by random candidate solutions including, although unnecessary, *a-priori* parameters, which may lead to a faster convergence (Ng, 1995). In a *population of individuals*, an EA conducts multiple searches in parallel by effective exchange of co-ordinate information (parameters) through crossover. At each stage of the evolution, the parameter values are altered in random (without directional guidance) by crossover and mutation. Then the performances (or *fitness*) of all candidate parameter sets are evaluated and the whole *generation* is guided *a-posteriori* to evolve in a “survival-of-the-fittest” manner. Hence superior parameter sets

would receive more attention for replication and refinement from generation to generation according to the Schema Theorem (Goldberg, 1989). The basic operation of a simple GA is shown below:

- Make initial population;
- Evaluate the fitness of all initial chromosomes;
- REPEAT
  - Reproduce children from parents;
  - Apply crossover and mutation to some chromosomes;
  - Extend the population with the children;
  - Select fittest chromosomes of the extended population for next cycle;
- UNTIL satisfactory result found or maximum generation number reached;
- Decode the best chromosome found.

To illustrate this, Figure 2.1 shows a GA example based on decimal coding.

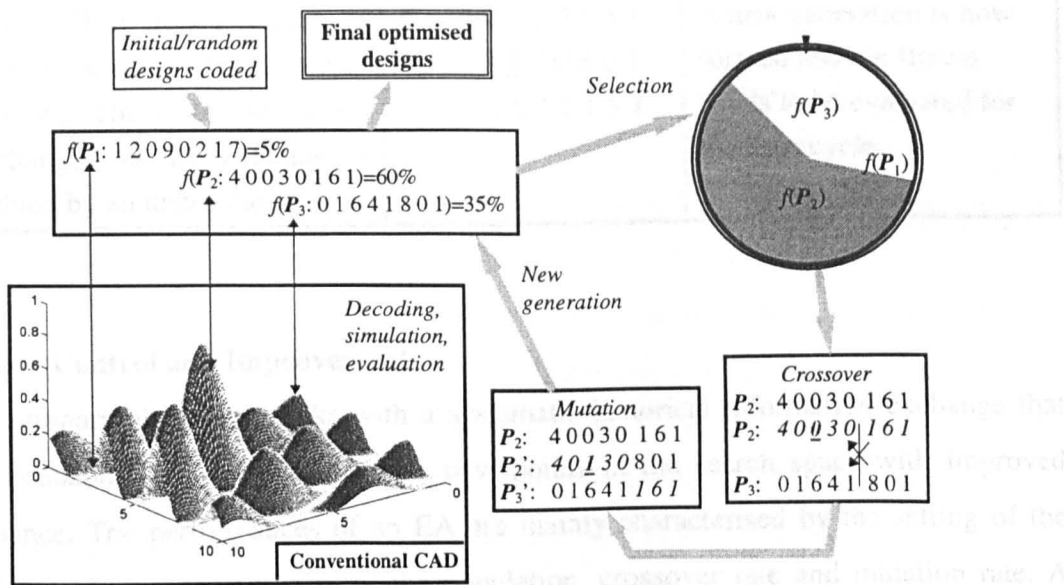



Figure 2.1 A GA transforming conventional CAD to “computer-automated design”

The operation details of Figure 2.1 may be explained in the following table:

Operations	Chromosomes	Fitness
<b>Initial population:</b> Example of coded parameter sets forming an initial population with size 3. The performance of each parameter set is simulated and then assigned a fitness.	$P_1$ : 1 2 0 9 0 2 1 7 $P_2$ : 4 0 0 3 0 1 6 1 $P_3$ : 0 1 6 4 1 8 0 1	$f(P_1) = 5 \%$ $f(P_2) = 60 \%$ $f(P_3) = 35 \%$ (N.B. The above fitness values are examples)
<b>Reproduction:</b> A simple scheme is to allow the chromosomes to reproduce off-spring according to their respective fitness. Thus $P_1$ has low probability of producing children, $P_2$ has a probability of producing two and $P_3$ one.	$P_2$ : 4 0 0 3 0 1 6 1 $P_2$ : 4 0 0 3 0 <u>1</u> 6 1  $P_3$ : 0 1 6 4 1 <u>8</u> 0 1	Evolution in progress (No need to re-calculate fitness here).
<b>Crossover:</b> Some portion of a pair of chromosomes is exchanged at the dotted position randomly specified.	$P_2$ : 4 0 0 3 0 1 6 1 $P_2'$ : 4 0 0 3 0 <u>8</u> 0 1 $P_3'$ : 0 1 6 4 1 <u>1</u> 6 1	No fitness calculations needed here.
<b>Mutation:</b> The binary values of some genes of some chromosomes are inverted. The value which has been changed as an example is highlighted by an underline.	$P_2$ : 4 0 0 3 0 1 6 1 $P_2''$ : 4 0 <u>1</u> 3 0 8 0 1 $P_3'$ : 0 1 6 4 1 1 6 1	A new generation is now formed and the fitness needs to be evaluated for the next cycle.

Algorithm Control and Improvement

An evolutionary algorithm works with a systematic historical information exchange that utilises probabilistic decisions to locate new points in the search space with improved performance. The performances of an EA are mainly characterised by the setting of the *control parameters*, namely, size of the population, crossover rate and mutation rate. A larger population size has the ability to avoid pre-mature convergence of the EA. However, it substantially requires more computational time in every generation. Although increasing crossover and mutation rate can introduce more new genetic materials in the population, a large crossover and mutation rate can disrupt the good chromosomes or transform the EA into a random search. The setting of these control parameters may depend on user's

experience and prior knowledge about the problem on hand. This is a drawback of evolutionary algorithms. However, a common choice of the population size may be set to about 10 times the complexity of the solution space and the crossover and mutation rates are usually set at 50~80% and 0.5%~2% of the population size, respectively.

Various work on further enhancements of the evolutionary algorithms originally developed by Holland (1975) and later Goldberg (1989) has been widely reported. Tournament, rank based and Boltzmann selection schemes (Baker, 1985; Sirag and Weisser, 1987; Srinivas and Patnaik, 1994) have been proposed to replace standard roulette-wheel selection for better diversity and efficiency. Extensions of a single-point crossover to two-point, multiple-point or uniform crossover have also been reported by Spears and DeJong (1991). Adaptive mutation and multiple range decoding schemes have been proposed (Ng, 1995), which need less prior experience in fixing the mutation rate and parameter range. *Generation gap* hypothesis (Grefenstette, 1986) was proposed to let the parents and children coexist in a same population and allows good genetic materials to be kept. Some of these techniques are detailed in the following sections.

## **2.2 Non-Binary Coding Schemes for System Parameters and Structures**

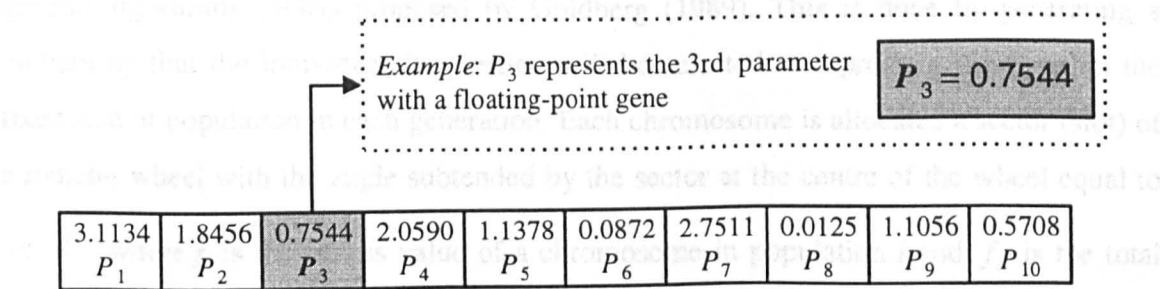
Classical genetic algorithms adopt a binary coding scheme to allow information to be coded in the chromosome and thus enable easy information exchange during the crossover process. A non-binary integer coding scheme was introduced by Homaifar and McCormick (1992) to reduce the chromosome length and the “Hamming Cliffs” effect encountered in a binary coding scheme. The “Hamming Cliffs” is defined as the largest Hamming distance between the binary codes of adjacent integers. For example, the codes 01111 and 10000 are the binary representation of 15 and 16, respectively, and have a Hamming distance of 5. For the GA to change the code of 15 to 16, it requires to change all the bits in 01111, which is obviously not very practicable. Although “Gray Codes” can also be used to reduce the Hamming Cliff effect (for binary coding), this is more complex in terms of decoding and in re-arranging the coding ranges for increased accuracy. Base-7 and decimal coding studied by Ng (1995) reduces the chromosome length and hence the Hamming distance. All base-2 (binary) to Base-10 (decimal) coding schemes can be represented by decimal integers

$P_i \in \{0, \dots, b-1\}$ , where  $b$  is the base value of the coding. Such a coding scheme for parameters of a control system involves an integer chromosome with its decimal value mapped as (Ng, 1995; Tan *et al.*, 1995):

$$D = D_{\min} + \frac{P_n b^n + P_{n-1} b^{n-1} + \dots + P_0 b^0}{b^n - 1} (D_{\max} - D_{\min}) \tag{2.1}$$

where  $D_{\min}$  and  $D_{\max}$  represent the range of the parameter values;  $D \in [D_{\min}, D_{\max}]$  is the real value being coded and  $n$  the number of digits which dictates the compromise between accuracy and speed in the search. The resolution of the decimal coding can easily be rearranged to be coarse initially and then be gradually refined as the population converges. A multiple decoding scheme (Ng, 1995; Tan *et al.*, 1995) with adaptive selection of searching range is adopted in this thesis. For example, each parameter is encoded by 4 digits, 3 of which represent the number of quantified values of the parameters within a given range selection. The extra digit is used as a “control gene” to search for an appropriate range of that parameter. The scheme benefits the EAs to have finer resolution and easy setting of the parameter range.

This non-binary coding scheme can be extended to floating-point coding (Li, 1996). For example, each parameter can be represented by a floating-point value, where several floating-point genes are combined to form a chromosome. Figure 2.2 illustrates the scheme for an EA with 10 parameters and with a searching domain between 0 and  $\pi$ . It can be seen that the third floating-point gene directly represents the value of the 3<sup>rd</sup> parameter.



A chromosome with 10 floating-point parameters

Figure 2.2 Floating-point coding scheme for  $P_{1,\dots,10} \in [0, \pi]$

No coding of chromosomes is used in “evolution strategy” or “evolutionary programming”. This is only feasible if the system structure in an engineering problem is fixed and only the parameters need to be found. The use of an additional “*control gene*” in decimal coding can provide a dimension for logic switching between possible parameter ranges and system structures (Li *et al.*, 1997). However, there are still limitations in efficiency and ranges of the parameters. This thesis has proposed another coding scheme in Section 2.7.3, the one-integer-one parameter coding, which overcomes these problems encountered in the decimal coding scheme.

## 2.3 Genetic Operators

### 2.3.1 Selection and Reproduction Schemes

Reproduction is used once the *initial population* involving, usually, a fixed number of chromosomes representing candidate designs are formed. In the reproduction process, a new generation of population is formed by randomly selecting individuals from an existing generation, according to their fitness, to breed. This fittest test is accomplished by adopting a selection scheme in which higher fitness individuals are being selected for contributing off-springs in the next generation.

#### 2.3.1.1 Roulette Wheel Selection

One of the standard selection method is the *roulette wheel selection* scheme in the simple genetic algorithms (SGA) proposed by Goldberg (1989). This is done by generating a probability that the individual in question will be selected to reproduce itself within the fixed size of population in each generation. Each chromosome is allocated a sector (slot) of a roulette wheel with the angle subtended by the sector at the centre of the wheel equal to  $2\pi \frac{f_i}{\hat{f}_i}$ , where  $f_i$  is the fitness value of a chromosome in population  $i$  and  $\hat{f}_i$  is the total fitness value of the population. A chromosome is selected for reproduction if a random generated number in the range of 0 to  $2\pi$  falls in the sector corresponding to the chromosome. The algorithm selects chromosomes in this fashion until it has generated the

entire population of the next generation. Although this selection scheme is easy to implement, several relatively high fitness individuals are always being selected in each roulette spin and dominating the whole reproduction process, which could lead to a premature convergence in the evolution. A different problem also arises in the later stages of the evolution when the population has converged and the variance in fitness becomes small. In this circumstance, the selection will fail to identify two chromosomes with small variance in fitness as they occupy almost similar sector size. The problems can be overcome by using scaling mechanisms or other selection schemes such as rank-based (Baker, 1985) or tournament selection schemes (Srinivas and Patnaik, 1994).

2.3.1.2 Rank-Based Selection

In ranking selection (Baker, 1985; Goldberg, 1989), population is sorted according to objective function value. Individuals are then assigned an offspring count using a predefined function. This approach provides a consistent means for offspring allocation and avoids the scaling problems encountered in the roulette wheel selection. Two types of ranking, arithmetic and geometric series ranking schemes are illustrated in Figure 2.3. This ranking cost-assignment procedure will be incorporated in the Pareto optimality multi-objective optimisation studied later in Section 2.6, in which the population is ranked on the basis of non-domination.

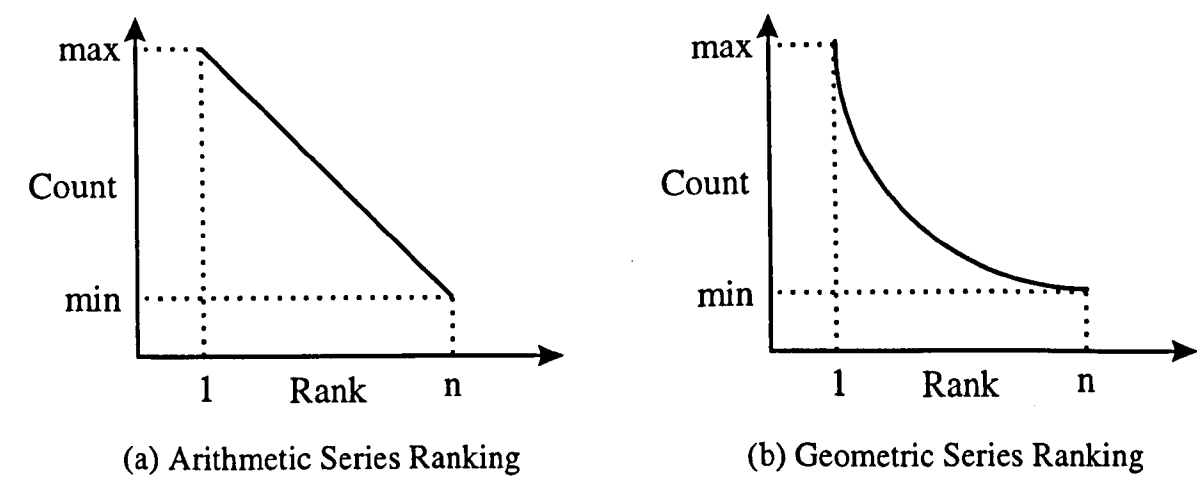


Figure 2.3 Rank-based selection scheme. (a) Arithmetic Series Ranking; (b) Geometric Series Ranking

2.3.1.3 Tournament Selection

In tournament selection as shown in Figure 2.4, the individuals are divided into subgroups and individuals with the best fitness among the subgroups are selected for reproduction. In deterministic tournament selection, the choice is made with probability 1, where stochastic tournament selection made with a probability usually less than 1. The subgroups can be of any size within the population. However, a usual choice is two or three for good diversity and preventing premature convergence of the EA.

A tournament selection scheme is shown in Figure 2.4. It has the following advantages over standard roulette wheel selection criteria (Srinivas and Patnaik, 1994):

- Deterministic and no scaling of fitness is required;
- Tournament size can vary
- Good diversity and efficiency

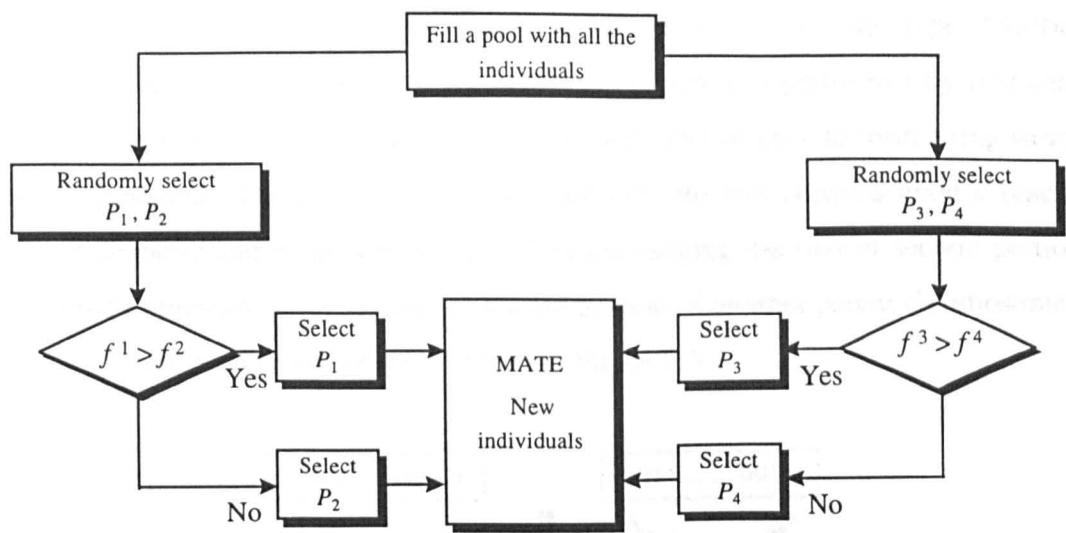


Figure 2.4 Tournament selection scheme with subgroups of size two



2.3.2 Crossover

Reproduction is in fact a selection process which chooses two parents for mating and does not generate novel individuals in the population. Therefore, as in natural sexual reproduction, the crossover operator is used to produce off-spring that are different from their parents but that inherit their parents' genetic material. Under this operator, a selected chromosome is split into two parts and recombining with another selected chromosome which has also been split at the same crossover point. An example of this crossover operation is illustrated in Figure 2.1. In addition to the single-point crossover, multiple random points or uniform crossover have been proposed and shown to be better than the standard single-point crossover (Spears and DeJong, 1991).

Based on the “Choice Theorem”, Zhang (1995) has reported that the best crossover point within a chromosome of  $N$  genes is  $N/2$ , provided the co-variances of parent chromosomes are uniformly distributed. Obviously, it is not appropriate to fix the crossover point on the centre of a chromosome, since it will lead to problem of chromosomes stagnation as the population evolves. Due to this, Zhang (1995) proposed a ring type “Sufficient Exchanging” optimal uniform crossover in which crossover is performed by first joining together the first gene of the parent chromosomes with its last gene to form a ring structure of the chromosome. Then, the chromosomes are cut into two portions upon a randomly generated diameter and crossover is realised by exchanging the first or second portion of one parent chromosome with the first or second portion of another parent chromosome. An operation of the ring type crossover is shown in Figure 2.5.

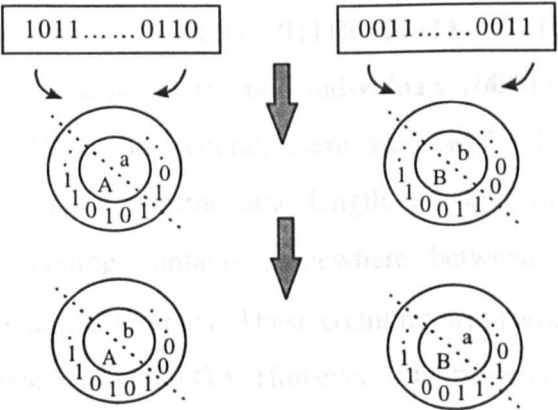


Figure 2.5 An illustration of ring type crossover

### 2.3.3 Mutation

Mutation plays a secondary role in an EA to alter the value of a gene at a random position on the chromosome string, discovering new or restoring lost genetic material. This serves to keep the diversity in the population and searches the neighbouring solution space, leading to an optimal answer. In a binary EA, the mutated genes are randomly selected and subjected to inversion of its value as illustrated in Figure 2.1. Decimal coding EA can perform the mutation operation by changing the value of a gene randomly or to its adjacent value (Ng, 1995). An adaptive mutation scheme which varies the mutation rate upon chromosome co-variance as in an evolution strategy (Li, 1996) has also been reported by Ng (1995). The mutation operator can also be realised by the Boltzmann learning technique (Tan *et al.*, 1995) which will be discussed later in Section 2.8.

## 2.4 Evolutionary Building Blocks and the Schema Theorem

The Schema Theorem developed by Holland (1975) offer a better understanding of the processing power of convergence of an EA. Although the framework of the Schema Theorem is establish upon the genetic operations of a GA, it provides a key explanation to the fundamental theory of other evolutionary algorithms. A schema is defined as a similarity template describing a subset of chromosomes with similarities at certain chromosome position. For example, in binary bit chromosome representation, a schema matches a chromosome if at every location in the schema a '1' matches a '1', a '0' matches a '0', or a '\*' ("don't care") matches either. Using this notation, the schema \*111\* describes a subset with four individuals {01110, 01111, 11110, 11111}. As another example, 0\*\*\*0 describes a subset with eight individuals {00000, 00010, 00100, 00110, 01000, 01010, 01100, 01110}. In general, there are  $(k+1)^l$  schemata for alphabets of cardinality 'k' and l being the chromosome length. As a result, a population with n individuals and binary coding contains somewhere between  $3^l$  and  $n \cdot 3^l$  schemata, depending upon the population diversity. These counting arguments provide an estimation of information being processed by the GA. However, it is important to distinguish between different types of schemata and thus the following two terminology are defined by Goldberg (1989). The order of a schema H, denoted by  $O(H)$ , is the number of fixed

positions within that schema. For example, the order of the schema 011\*1\*\* is 4. The defining length of a schema  $H$ , denoted by  $D(H)$ , is the distance between the outer-most fixed position. In the example above, the defining length of the schema is 4. The “*implicit parallelism*” by Holland (1975) estimates the number of schemata processed in a population of  $n$  chromosomes is equal to  $O(n^3)$ . This yields a processing power less dependent on the length of the chromosome (i.e., dimension of the problem) and gives the GA computational advantage that traditional deterministic search methods cannot match.

Apart from classifying the chromosome similarities, schemata and their properties provide the basic means for analysing the net effect of reproduction and other genetic operators in a GA. The combined effect of reproduction, crossover, and mutation on schemata contained within a population of chromosomes from generation  $g$  to generation  $g+1$  can be expressed in the following equation (Goldberg, 1989; Holland, 1975):

$$m(H, g+1) \geq m(H, g) \cdot \frac{f(H)}{\bar{f}} \left[ 1 - p_c \frac{D(H)}{l-1} - p_m O(H) \right] \quad (2.2)$$

where  $m$  is the number of chromosomes belonging to schema  $H$ ;  $\bar{f}(H)$  is the average fitness of all chromosomes;  $f(H)$  is an estimate of the average fitness of those chromosomes which are members of schema  $H$ ;  $p_c$  and  $p_m$  are the crossover rate and mutation rate, respectively. This means that highly fit, low order and short defining length schema (known as *building blocks*) are increased rapidly in the population. These building blocks are sampled, recombined, and resampled to form chromosomes of potentially higher fitness. This has formed the fundamental theory of the genetic algorithm (Goldberg, 1989). Other than the binary representation, the building block hypothesis is also applicable to other non-binary coding schemes (Ng, 1995).

## 2.5 Other Techniques in EA Operations

### 2.5.1 Niche and Speciation

A niche is viewed as an organism’s environment and a species is a class of organism with common characteristics. This separation of the environment and the organisms exploiting that environment into different subsets by forcing similar individuals to share their resources is very common in nature (Goldberg, 1989). As illustrated in Figure 2.6, a simple EA without niching has been shown to converge to a single optimum even though multiple equal or unequal optimal solutions exist (Goldberg, 1989). To encourage the formation of niche and species in an EA, it is essential to introduce a control competition among different solutions near every locally optimal region and thus maintain a stable subpopulations at such optimal regions.

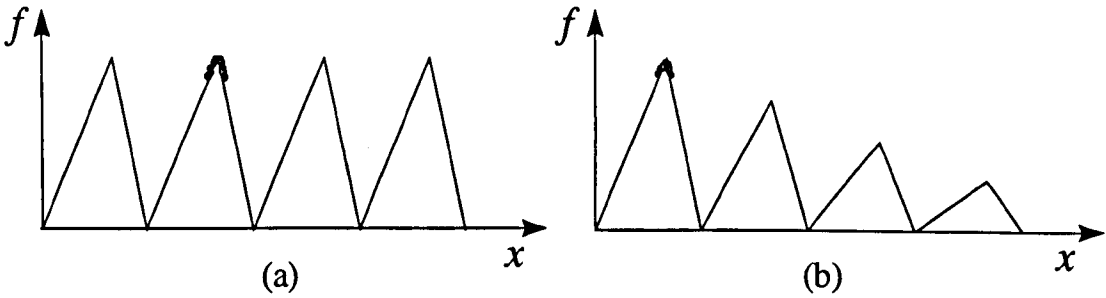


Figure 2.6 Performance of simple EA without niching (a) equal peaks and (b) unequal peaks

Niching in general is achieved using a sharing function that creates subdivisions of the environment by degrading an organism’s fitness  $f_i$  proportional to the number of other members in its neighbourhood (Goldberg, 1989; Krishnakumar and Satyadas, 1996), which is employed at the end of each generation. This degradation is obtained by dividing the fitness by a niche count  $m_i$  to find the shared fitness as given by

$$SharedFitness = \frac{f_i}{m_i} \tag{2.3}$$

The niche count is an estimate of how crowded is the neighbourhood (niche) of individual  $i$ . It is calculated over all individuals in the current population as given by

$$m_i = \sum_{j \in Pop} Sh(d_{ij}) \quad (2.4)$$

where  $d_{ij}$  is the distance between an organism  $x_i$  and its neighbour  $x_j$ , which is calculated by their proximity in the decoded parameter space (Krishnakumar and Satyadas, 1996). Given  $p$  parameters of unequal boundaries over a parameter range  $[x_{\min}, x_{\max}]$ ,

$$d_{ij} = \sqrt{\sum_{k=1}^p \frac{x_{k,i} - x_{k,j}}{(x_{k,\max} - x_{k,\min})^2}} \quad (2.5)$$

$Sh(d_{ij})$  is the sharing function. For each  $d_{ij}$ , the following power law sharing function  $Sh(d_{ij})$  may be used (Goldberg and Richardson, 1987)

$$Sh(d_{ij}) = \begin{cases} 1 - \left( \frac{d_{ij}}{\sigma_{share}} \right)^w & \text{if } d_{ij} < \sigma_{share} \\ 0 & \text{otherwise} \end{cases} \quad (2.6)$$

$\sigma_{share}$  is the limiting distance between the individuals to be shared. It is usually fixed by the user at some estimate of the minimal separation desired or expected between each niche in the solution space. It can be calculated as (Krishnakumar and Satyadas, 1996)

$$\sigma_{share} = 0.5n^{\left(\frac{1}{p}\right)} \quad (2.7)$$

where  $n$  is the assumed peaks in the solution space. The triangular sharing function (Goldberg and Richardson, 1987) with  $w = 1$  in Equation (2.6) is adopted in this thesis, since it allows equal degree of sharing between the neighbouring individuals. As illustrated in Figure 2.7, there is a fractional contribution sharing effect when  $0 < d_{ij} < \sigma_{share}$ . Two individuals will share a full portion with each other while  $d_{ij} = 0$ . When the individuals are separated  $> \sigma_{share}$  or far apart, they produce no effect on each other.

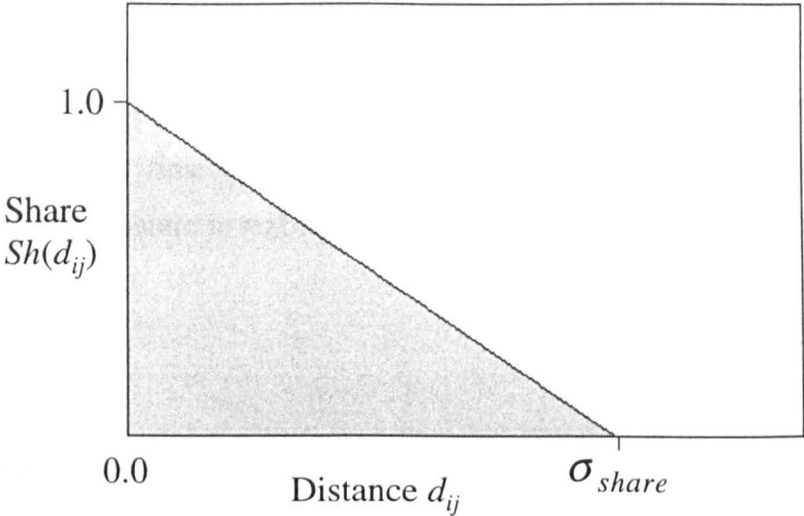


Figure 2.7 Triangular sharing function

2.5.2 Mating Restriction

Mating restriction (Fonseca and Fleming, 1993; Goldberg, 1989; Tan and Li, 1997b) tries to favour the mating of similar individuals, since arbitrary mating of individuals may result in a large number of unfit offspring. Mating restriction can be implemented in the same way as fitness sharing by introducing a corresponding  $\sigma_{mate}$  to specify how close individuals should be in order to mate (Fonseca and Fleming, 1993). The choice of  $\sigma_{mate} = \sigma_{share}$  is adopted in this thesis. After selection, one individual in the population is chosen and the population searched for a mate within the distance  $\sigma_{mate}$ . Mating is performed if such an individual can be found, otherwise a random individual is chosen for the mating.

2.5.3 Generation Gap

*Generation gap* hypothesis (Grefenstette, 1986) suggests that a percentage of good chromosomes in a generation should be passed to the next generation. This subset of the population goes through regular selection for mating purposes but is not altered going into the next generation. In contrast to the generational EA that replaces the entire population in each generation by their off-spring, a steady state EA (Grefenstette, 1986) retains a  $\eta$

percentage of the population selected based on their fitness into the next generation. The idea is to keep good genetic materials to be passed on to the next generation, leading to reservation of potentially better solutions and thus improves the convergence rate and saves the overall computation time by reducing the number of function evaluations. This approach is in fact very nature in real world to allow the parents and off-springs to coexist in the same population.

#### 2.5.4 Parallelism

EAs are inherently parallel in the sense that each chromosome update is largely independent of the others. In a parallel EA model, the full population exists in a distributed form, either multiple independent sub-populations exist, or there is one population but each population individuals interacts only with a limited set of neighbours. A simple way of parallel implementation of a traditional sequential EA model is to parallelise the genetic operators and the function evaluations associated in a generation loop. There are a few advantages of parallel EAs over sequential EAs:

- Reduced overhead computational time;
- It is more realistic model of species in nature than a single large population;
- It maintains more diverse sub-populations by exchanging genetic material between sub-populations, which avoids premature convergence of the EA.

Parallel EA can be classified according to the granularity of the distributed population, *coarse grained* vs. *fine grained* (Gordon and Whitley, 1993). In a coarse-grained (or island model) parallel EA, the population is divided into several sub-populations, each of which runs a traditional EA independently and in parallel on its own sub-population. Occasionally, fitter strings migrate from one sub-population to another. In a fine-grained parallel EA, a single population is divided so that a single string is assigned to each processor. Processors selected from, crossover with, and replace only strings in their neighbourhood. Since neighbourhoods overlap, fitter chromosomes will migrate throughout the population.

## 2.6 Multi-Objective Optimisation

### The Problem

Multi-objective (MO) optimisation seeks to optimise a vector of non-commensurable and often competing objectives or cost functions, viz. it tends to find an optimal parameter set  $P$  for

$$\min_{P \in \Phi} F(P) \quad (2.8)$$

where  $P = \{p_1, p_2, \dots, p_n\}$  and  $\Phi$  defines the set of candidate parameter variables;  $F: \mathcal{R}^n \rightarrow \mathcal{R}^m$ ,  $F(P) = \{f_1(P), f_2(P), \dots, f_m(P)\}$  for  $m$  objectives. The solution to the problem is a family of points known as a Pareto optimal set, where each of the objective components of any point along the Pareto front can only be improved by degrading at least one of its other objective components. Conventional MO techniques include the methods of inequalities, goal attainment or weighted sum. However, these approaches require a set of precise settings of weights or goals prior to optimisation, which are usually not well manageable or understood (Grace, 1992; Osyczka, 1984). If the solution produced is not satisfactory, the weights or goals must be changed and the optimisation process is required to repeat again.

Evolutionary algorithms have been recognised to be well-suited to multi-objective optimisation (Goldberg, 1989; Fonseca and Fleming, 1993). Unlike conventional methods that linearly combine multiple attributes to form a composite scalar objective function, a multi-objective evolutionary algorithm (MOEA) incorporates the concept of Pareto's domination to evolve a family of non-dominated solutions at multiple points along the Pareto optimal frontier simultaneously and efficiently. By combining the Pareto dominance with partial preference information in the form of a priority vector, each of the individual components in the cost function can have different priorities or preferences to guide the optimisation from individual specifications rather than pre-weighting the cost function (Fonseca and Fleming, 1993; Tan and Li, 1997b).



Pareto Optimality

Pareto-based fitness assignment was first proposed by Goldberg (1989), as a means of assigning equal probability of reproduction to all non-dominated individuals in the population. The method assigns rank 1 to the non-dominated individuals and removing them from contention, then finding a new set of non-dominated individuals, ranked 2, and so forth. Tournament selection based on Pareto dominance was later proposed by Horn *et al.*, (1994). In addition to the individuals competing in each tournament, a number of other individuals in the population were used to help determine whether the competitors were dominated or not. Sharing is used to determine the selection if both competitors were either dominated or non-dominated.

Fonseca and Fleming (1993) have developed a slightly different scheme to rank the individuals, which is illustrated in Figure 2.8 with two objectives  $f_1$  and  $f_2$  to be minimised. They assign the same smallest cost for all non-dominated individuals, while the dominated individuals are inverse ranked according to how many individuals in the population dominate them. So, an individual's rank in a population can be given by  $rank(p_i) = 1 + k_i$ , where  $k_i$  is the number of individuals that dominate the individual  $p_i$  in the current population. All the non-dominated individuals will be assigned rank 1. Their algorithm proceeds by sorting the population and assigning the fitness by interpolating before averaging it between individuals with the same multi-objective rank.

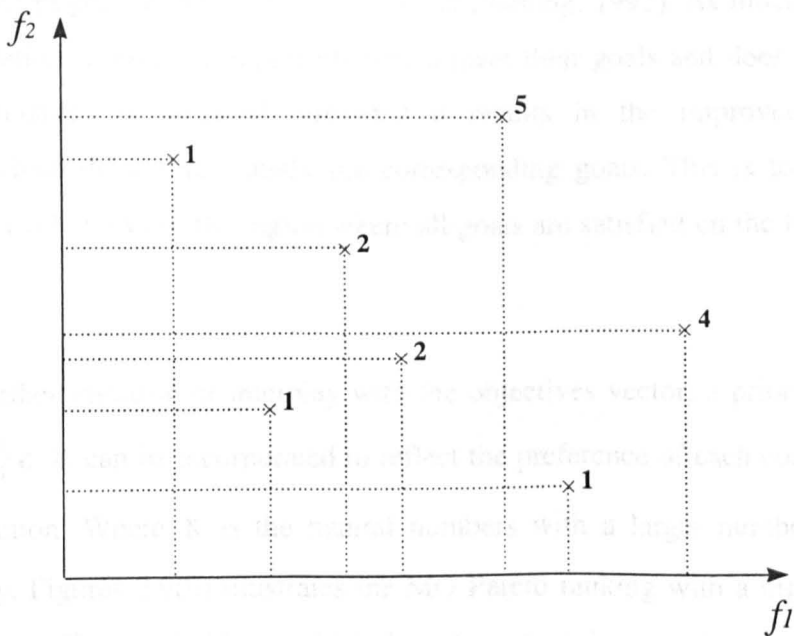


Figure 2.8 Multi-objective Pareto ranking

### Goal and Priority

Goal and priority information is often naturally available from the problem formulation, although they are not necessarily utilised in a strict sense. The cost assignment method described earlier can be modified to accommodate goal information (Fonseca and Fleming, 1993; Tan and Li, 1997b) in a similar way to that used by conventional goal attainment method (Grace, 1992). The method of goal attainment entails the construction of a set of *goal* values for the objective functions. It converts the multi-objective optimisation problem in Equation (2.8) into the following nonlinear programming problem by finding,

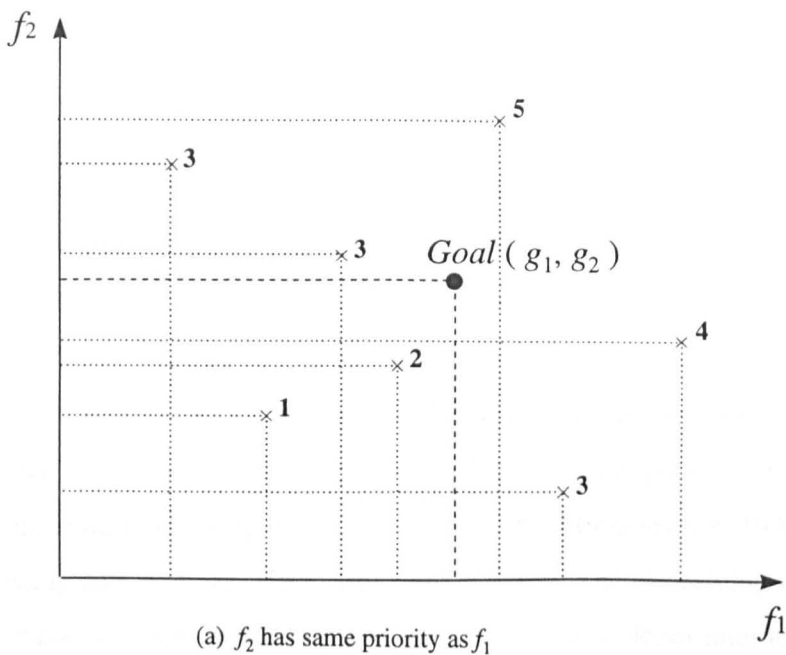
$$\min_{P \in \Phi} \lambda \tag{2.9}$$

such that

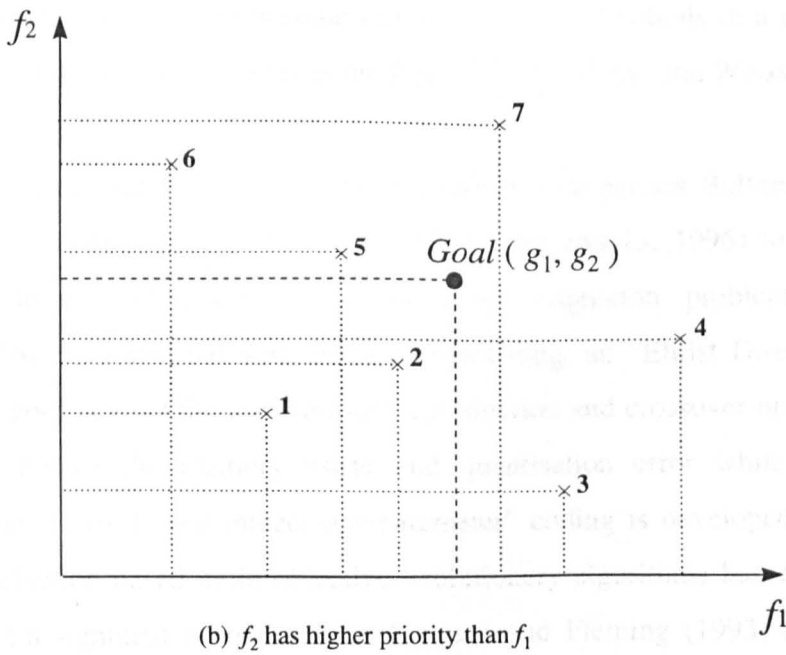
$$f_i - w_i \lambda \leq g_i \tag{2.10}$$

where  $g_i$  are goals for the design objectives  $f_i$ , and  $w_i \geq 0$  are weights. The minimisation of the scalar  $\lambda$  leads to the finding of a non-dominated solution which under- or over-attains the specified goals to a degree represented by the quantities  $w_i \lambda$ . By altering the way in which individuals are compared with one another, the MO ranking procedure can be extended to accept goal information (Fonseca and Fleming, 1993). As illustrated in Figure 2.9(a), degradation in vector components which meet their goals and does not go beyond the goal boundaries is accepted, provided it results in the improvement of other components which do not yet satisfy the corresponding goals. This is to encourage all individuals to evolve toward the region where all goals are satisfied on the Pareto trade-off surface.

In order to further visualise or interplay with the objectives vector, a priority vector  $\mathbf{G}_i = \{g_1, g_2, \dots, g_m\} \in \mathbb{N}$  can be incorporated to reflect the preference of each component in the objective function. Where  $\mathbb{N}$  is the natural numbers with a larger number represents a higher priority. Figures 2.9(b) illustrates the MO Pareto ranking with a higher preference upon objective  $f_2$ . Here, individuals which do not meet goal  $g_2$  are the worst, independent of their “theoretical” performance according to  $f_1$ . Once  $g_2$  is met,  $f_1$  is used for ranking.



(a)  $f_2$  has same priority as  $f_1$



(b)  $f_2$  has higher priority than  $f_1$

Figure 2.9 Multi-objective ranking with goal and priority values

## 2.7 Improving EA Operations

### 2.7.1 Existing EA Problems

Evolutionary techniques are based on a multi-point *a-posteriori* trial-and-error-process, guided by the Darwinian-Wallace principle of “survival-of-the-fittest”. The search power of an EA mainly lies in its crossover and mutation operators, which provide the diversity of candidate “species” and varying search domains. Despite recent techniques developed to improve the performance of an EA as discussed in the previous sections, it is well-known that existing EAs are weak in local exploration and are thus poor in finding the exact optima at each generation (Kwong *et al.*, 1995; Li, 1996; Michalewicz, 1994; Renders and Bersini, 1994; Sirag and Weisser, 1987; Tan and Li, 1996). The underlying reason of this is that, in an EA, there is a lack of “biological diversity” resulting from interactions with, and thus direct learning from the evolution environment. In addition, as mutation rate is usually set at very low to avoid the EAs become random search, individuals in a population may become more and more homogeneous as the EAs converge (Sirag and Weisser, 1987).

In this Chapter, a hybrid EA will be reported, which incorporates Boltzmann’s learning capabilities in local fine-tuning (Tan *et al.*, 1995; Tan and Li, 1996) to overcome the weakness in local exploration and chromosome stagnation problems existing in generational EAs. Incorporated with Boltzmann learning, an “Elitist Direct Inheritance” technique that goes beyond the conventional reproduction and crossover operations is also proposed. To reduce the memory usage and quantisation error while having faster processing time, a novel “one-integer-one-parameter” coding is developed in this thesis. Tournament selection based multi-objective evolutionary algorithms based on the Pareto optimality cost-assignment proposed from Fonseca and Fleming (1993) is also studied. This approach avoids the need of chromosome sorting, interpolating and fitness averaging in every generation and thus saves the computational time. Parallelism of the EA using a 64-processor Parsytec transputers in a 2-D array is also implemented here.

## 2.7.2 Fitness Functions for Control Engineering Problems

### Example of Fitness Function for System Identification

The fitness function in an EA is similar to the inverse of a cost-function in other optimisation techniques. The choice of the fitness function is crucial. It reflects the objectives or specifications of the application, which directs the searching process and also affects the rate of convergence. In the problem of system modelling, for example, parameters are usually chosen in such a way that the average of the squares of the differences between the actually observed data and the computed values is minimal. Thus, a simple fitness function can be given by:

$$f = \frac{1}{1 + \sqrt{\sum_{i=1}^N [e_i^2] N^{-1}}} \quad (2.11)$$

where  $i$  is the time or frequency index in the simulation;  $N$  is the window size;  $e_i$  represents the error between the actual and the simulated values. The main reason why this minimum least squares criterion is so popular is that the theory of least squares optimisation has been well developed over the last few decades. Different criteria like quadratic errors ( $L_2$  norm) or application-oriented errors (such as  $L_\infty$  norm) may also be used and have been employed for the system identification and modelling problems studied later in Chapters 3 and 4. Apart from these, minimum absolute errors ( $L_1$  norm) may also be incorporated in an EA (Sharman *et al.*, 1995).

### Example of Fitness Function for Controller Design

In control system designs, a simple fitness function that reflects small steady-state errors, a short rise time, low oscillations, low overshoots and a good stability may be given by:

$$f = \frac{1}{\sum_{i=1}^N \{e_i^2 + \Delta e_i^2\} i} \quad (2.12)$$

where  $e_i$  is the error between the command and response, and  $\Delta e_i$  the amount of change of  $e_i$  between two adjacent time indices, which may be distinctively weighted to further suppress oscillations. In Equation (2.12),  $N$  is the total time steps chosen to ensure that the simulation reaches steady-state. It varies with the system time-constant and the sampling rate. Clearly, this fitness function may be used under the condition of other system constraints such as output voltage limitation. Other customer specifications and some robustness criteria can also be included in this fitness function by either arithmetic or logic operations. Since the EA is based on fitness evaluation, no differentiation of this performance index is needed and thus discontinuous fitness is allowed.

However, the above fitness function of Equation (2.12) may not be adequate in control system design, particularly for multi-objective optimisation problems. Alternately, the concept of Pareto's domination and goal attaining techniques can be incorporated in the EAs to define a "fitness function" without the need to linearly combine these multiple attributes to form a composite scalar objective function. Techniques for solving multi-objective optimisation using the EAs will be addressed in Section 2.9 and details regarding fitness function selection for controller design in Chapter 5.

### **2.7.3 One-Integer-One-Parameter Coding Scheme**

A novel one-integer-one-parameter coding is developed in this thesis, which takes the advantage of faster processing speed in integer data type representation of the software developing tools. A 16-bit integer data type has a value between -32768 and +32767 and can be used to represent one parameter in a candidate model. For example, suppose the one-integer-one-parameter coding is used for an EA with 10 parameters and a searching domain between  $-\pi$  and  $\pi$ . If a resolution of 0.0001 is needed, then the largest decoded parameter values would be  $\frac{\pm 32767}{10000} = \pm 3.2767$ . In this case, an overflow checking is carried out such that if any decoded parameters are out of the range of  $\pm \pi$ , the parameters will be assigned to the value of  $\pm \pi$ . In a uniform LTI control system design (ULTIC) as developed later in Chapter 5, such an overflow/underflow checking is found particularly

effective in finding integrators or low order controllers if any parameters are assigned a value of 0 whenever it is out of the searching range. Different integer data types, such as long integers or unsigned integers, can also be used if a larger resolution is required.

2.7.4 Direct Inheritance as Selection and Crossover

Proportionate and elitist “Direct Inheritance” (DI) techniques that go beyond the conventional selection and crossover operations are proposed in this thesis. They apply the similar concept of “survival-of-the-fittest” by direct uni-directional transferring genes from a fitter chromosome to a weaker chromosome. They ensure that only the fitter chromosomes are given the priority to “pass-on” their genes to the next generation in an evolution process. The techniques are performed by dividing parents into a subgroup of size two. Fitter chromosomes in the subgroups are selected for Boltzmann type Simulated Annealing (SA) learning for an improved EA that will be developed in Section 2.8. Within a subgroup, genes are randomly copied from a fitter chromosome to the corresponding genes in a weaker chromosome. In a proportionate DI technique as depicted in Figure 2.10, the number of genes transferred are depend on the fitness proportion between the two chromosomes A and B, i.e.,  $\lceil \frac{f_a}{f_a+f_b} \rceil$ , where  $f_a > f_b$  are the fitness of the two chromosomes, respectively. Thus, the fitter the chromosome-A related to chromosome-B, the more of its genes will be copied to chromosome-B.

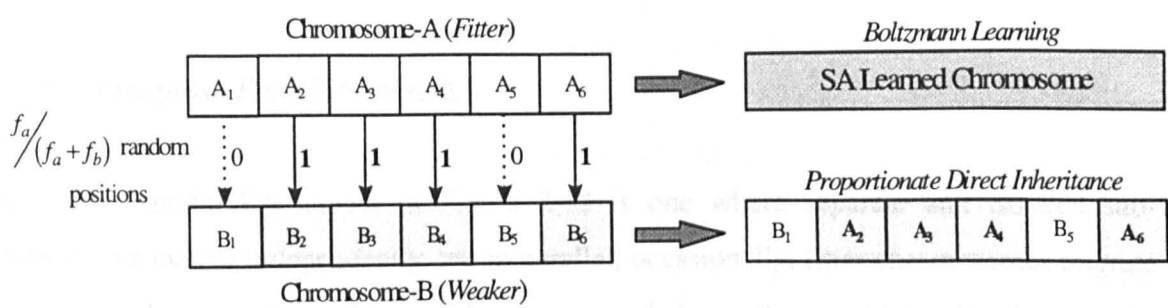


Figure 2.10 “Proportionate Direct Inheritance” operation

However, genes copying based upon fitness may not be appropriate since a linear scale fitness function is required to avoid bias inheritance. Inspired by the “Choice Theorem” (Zhang, 1995), a novel “Elitist Direct Inheritance” technique is proposed to overcome the deficiency encountered in a proportionate DI. As depicted in Figure 2.11, the elitist DI technique is performed by randomly copying  $\lceil N/2 \rceil$  genes from a fitter chromosome-A to a weaker chromosome-B, where  $N$  is the number of genes in a chromosome. This approach has encouraged an uniformly distributed co-variances of the parent chromosomes and fulfils the assumption that required by the “Choice Theorem”. In addition, elitist DI technique does not require the user setting of crossover rate and mutation rate and hence reduces the *control parameters* that characterise the performances of EAs. This elitist DI technique is used in the benchmark test studied in Section 2.8.2 and has shown to be better than a simple GA or SA.

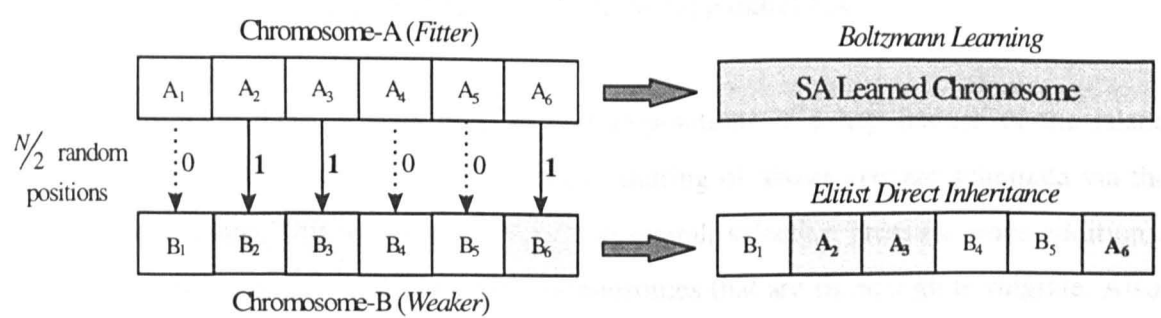


Figure 2.11 “Elitist Direct Inheritance” operation

2.7.5 Transputer Based Parallel EA

An island model EA shown in Figure 2.12 is one where separate and isolated sub-populations evolve independently and in parallel, occasionally, fitter chromosomes migrate between sub-populations. Many EA researches believe that multiple distributed sub-populations, with local rules and interactions, are a more realistic model of species in nature than a single large population (Goldberg, 1989; Gordon and Whitley, 1993). Gordon and Whitley (1993) compare eight different parallel EAs and a version of Goldberg’s Simple Genetic Algorithm (Goldberg, 1989) on several function optimisation test



problems. Among their conclusions is that island models perform well, particularly on the hardest problems in their test suite.

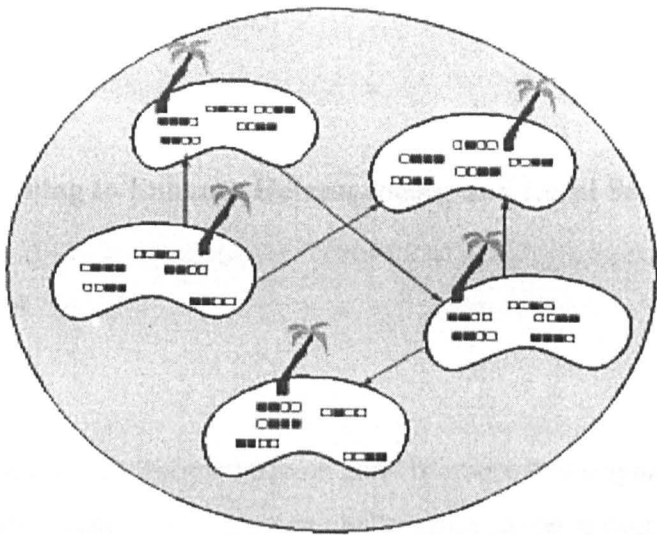


Figure 2.12 An island model parallel EA

The migration of chromosomes between sub-populations is a key feature of the island model EA. First, it allows the distribution and sharing of above average schemata via the strings that migrate. This serves to increase the overall selective pressure since additional reproductive trials are allocated to those chromosomes that are fit enough to migrate. Also, the introduction of migrant strings into the local population helps to maintain genetic diversity, since the migrant chromosomes arrives from a different sub-population which has evolved independently. If, however, too many copies of fit chromosomes migrate too often, the EA may suffer from premature convergence. It is possible that after a certain number of migration steps each sub-population contains a copy of the globally fittest individual, and copies of this chromosome (and only this chromosome) migrate between sub-populations (Gordon and Whitley, 1993; Li *et al.*, 1996b).

The 64-processor Parsytec transputer system was used for parallel implementation of the EA described above. All the simulation task was equally shared by up to 64 T8 transputers in a 2-D array. All the transputer including the host for communications and supervisory tasks are provided by a Parsytec SuperCluster. Parallel C is used under the PARIX (PARallel unIX) operating system that offers straightforward software-channels for inter-transputer communications. Each time a parallel run was made, all sub-populations were

randomly seeded. This was done by having processor zero get and broadcast the microsecond portion of the Unix `gettimeofday` system call. Each processor then added its processor id to the value returned by the Unix `gettimeofday` and used this unique value as its random number seed.

## **2.8 Boltzmann Learning to Enhance Heterogeneous and Local Search**

### **2.8.1 The Algorithm**

#### **Hillclimbing EA**

To improve the weakness in local exploration and chromosomes stagnation problems in an EA, it is intended to combine the “generational” optimisation power of crossover of an evolving population with the local fine learning individuals in each generation. “Positive mutations” resulting from learning can be implemented through trial-and-error again, requiring no derivative information. While a trial finds a new chromosome of a smaller cost by mutating a small number of genes, 5% ~ 10%, of an existing chromosome, it replaces the existing with the new. This trial may be refined several times as in a “hillclimbing” algorithm as illustrated in the flow chart shown in Figure 2.14 excluding the dotted decision block. This algorithm is similar to that presented by Renders and Bersini (1994) where Powell’s directional hillclimbing is used in addition to the existing mutation operation. Note that, if directional information can be available, it should be employed to direct the fine search for quicker local exploration.

#### **Boltzmann EA**

The hillclimbing technique refines search towards absolutely “improved” performance, which may be prone to false directions due to local minima or numerical inaccuracy (or noisy data). This may be overcome by replacing hillclimbing with the simulated annealing (SA) technique (Tan and Li, 1996), which asserts a probability of retaining possibly correct search directions. As shown in Figure 2.14 including the dotted block, here an existing chromosome may be replaced by a slightly inferior chromosome with a probability higher than the Boltzmann selection criterion given by the following decision making:

If

$$\exp\left[-\frac{J(C) - J(C_k)}{k_B T}\right] > \text{random}(0, 1)$$

then

move from energy level  $J(C_k)$  to  $J(C)$

Here the function  $\text{random}(0, 1)$  returns a random number valued between 0 and 1 with an uniform probability density of 1;  $k_B$  may be the Boltzmann's constant, but can be set to an artificial value in the annealing process. It is  $5 \times 10^{-6}$  in this thesis for a fine annealing decision coupled with the "annealing temperature"  $T$ ;  $C_k$  and  $C$  represent a current chromosome and its mutated version. In this scheme, the annealing temperature is initially set to  $T_{\text{ini}}$ . This is relatively large, in order to allow a wide neighbourhood mutations accommodating more possible directions for correct convergence. The annealing temperature  $T$  will decrease exponentially with the number of the annealing cycle  $j \in [1, j_{\text{max}}]$  at the rate of  $\beta^{j-1}$ , where  $\beta < 1$  is the annealing factor. The learning is getting tighter as  $T$  decreases. The final temperature  $T_{\text{final}}$  is determined by how tight the fine-tuning should be bounded at the end of the learning process. According to Figure 2.14, the final temperature is given by:

$$T_{\text{final}} \geq \beta^{j_{\text{max}}} T_{\text{ini}} \quad (2.13)$$

where

$$j_{\text{max}} = \left\lceil \log_{\beta} \left( \frac{T_{\text{final}}}{T_{\text{ini}}} \right) \right\rceil \quad (2.14)$$

In this thesis, a fast annealing scheme ( $\beta = 0.3$ ) is used to limit the total number of SA refinements of a chromosome in one EA generation constantly to  $j_{\text{max}} = 10$  for  $T_{\text{ini}} = 10^5$  and  $T_{\text{final}} = 1$ .

### Gradual Boltzmann EA

Clearly, learning at pre-converging stages are not as valued as that occurring at final stages of evolution. Gradual learning effort can be achieved by varying  $\beta$  according to, for example, the following linear or nonlinear type of schedules:

$$\beta(g) = \frac{g}{g_{\max}} \beta_0 \quad (2.15a)$$

$$\beta(g) = \beta_0 \left( 1 - e^{-\frac{g}{\tau}} \right) \quad (2.15b)$$

where  $g$  represents the number of generations evolved,  $g_{\max}$  is the maximum number of generations used as one of the means to terminate the evolution process;  $\tau$  is the time constant used to determine transient behaviour of the nonlinear learning and  $\beta_0 = 0.3$  the initial annealing factor. Figure 2.13 shows an example of the linear and nonlinear learning schedules for  $g_{\max} = 100$  and  $\tau = 0.1 g_{\max}$ . As the evolution converges, the chromosome co-variances in a population are also decreased. Thus, there is no need to use adaptive mutations (Ng, 1995) that determine the mutation rate by chromosome co-variances. Instead, indirectly increasing the mutation rate with the generation number as shown in Figure 2.13 requires less management time and suffices, as  $j_{\max}$  increases with  $\beta$  and with chromosome similarities. Further, the increased mutations are harmless to disturb the converging population, because the mutations are indirectly guided by local learning.

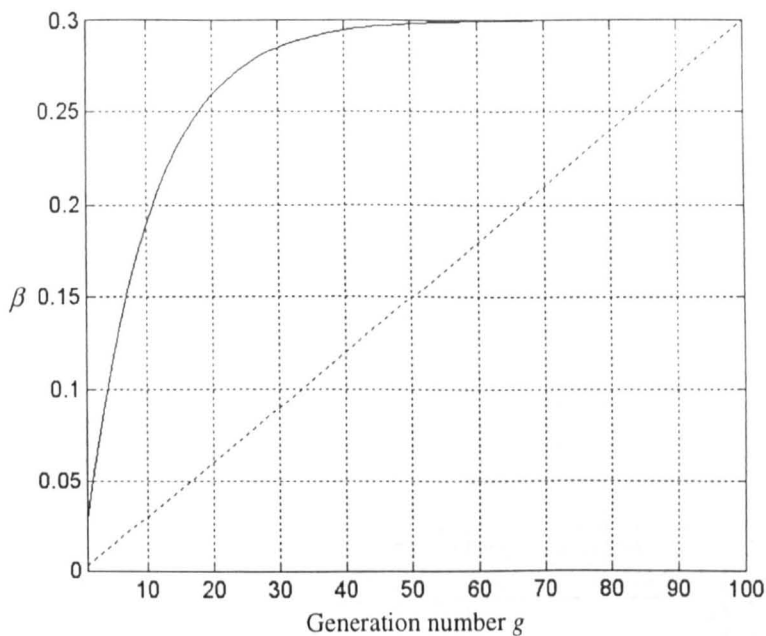


Figure 2.13 Linear ( - - - ) and nonlinear ( — ) Boltzmann learning schedules

In Figure 2.14, parents are divided into a subgroup of size two and 45% of fitter chromosomes within the subgroups are selected for Gradual Boltzmann learning. The learned springs (45% in total) will be mixed with the Elitist DI inherited springs (50% in total) and 5% of the parents ( $\eta = 5\%$  in generation gap maintenance) to form a new population. This improved hybrid evolutionary algorithm is employed in solving the engineering problems detailed later in Chapter 3, 4, 5 and 6 in this thesis.



Figure 2.14 Flow chart of an EA along "positive mutations" with a Boltzmann learning schedule

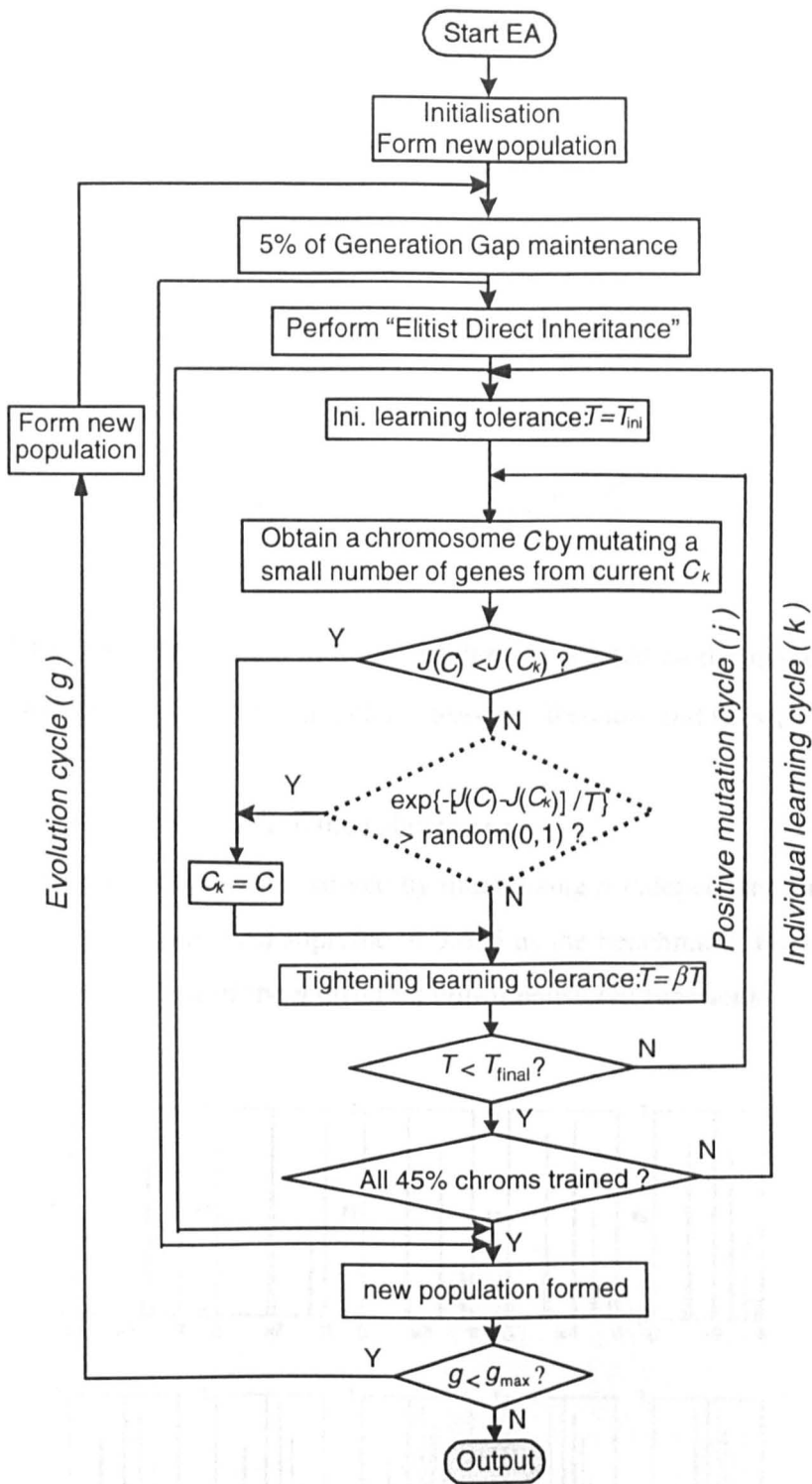


Figure 2.14 Flow chart of an EA using “positive mutations” with a Boltzmann learning schedule

2.8.2 Benchmark Test

A maximisation problem of a family of functions has been studied by Michalewicz (1994) and Renders and Bersini (1994) as “benchmark” to test different evolutionary algorithms. It is also to be used here to evaluate the proposed hybrid Boltzmann learning EA. The function is an amplitude modulated  $n$ -dimensional sine-wave of nonlinearly increasing frequency:

$$f(\mathbf{x}) = \sum_{i=1}^{10} f_i(x_i) = \sum_{i=1}^{10} \sin(x_i) \sin^{200}\left(\frac{ix_i^2}{\pi}\right) \tag{2.16}$$

where  $x_i \in [0, \pi]$ . The fitness used in the algorithms is defined as the maximum value of  $f(\mathbf{x})$ . The function  $f(\mathbf{x})$  has the following characteristics (Renders and Bersini, 1994):

- 1. There are 3628800 local optima in the solution space.
- 2. This optimisation problem can be solved by maximising  $n$  independent uni-dimensional functions, yielding a theoretical supreme of 9.655 as the benchmark. Figure 2.15 shows the exhaustive evaluations of the  $n$  different uni-dimensional functions.

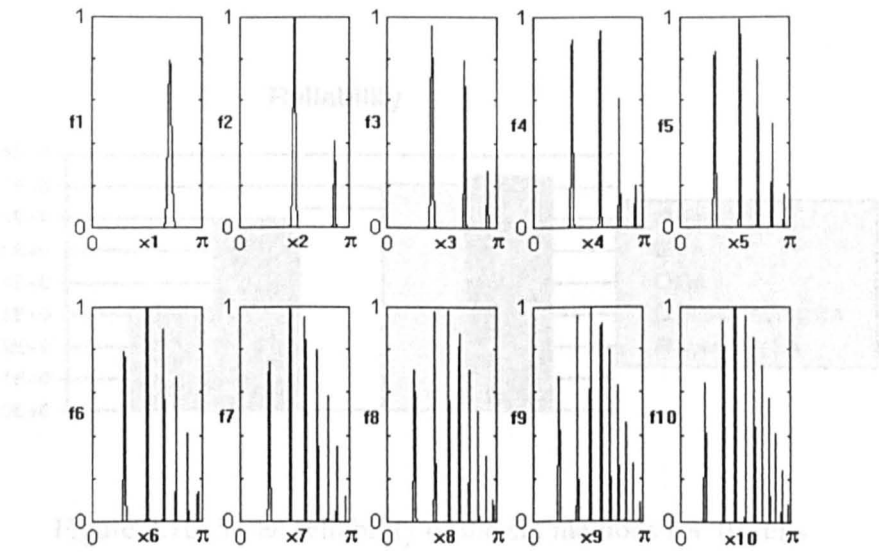


Figure 2.15 Exhaustive search solutions to the  $n$  independent uni-dimensional function optimisation problem

The following performance measures (Renders and Bersini, 1994) are used for comparisons in the benchmark tests:

- Reliability: This is indexed by the number of “correct” parameters  $x_i$ . A parameter  $x_i$  is considered as correct only if it is situated on the peak corresponding to the global maximum for this component.
- Number of evaluations: This is the total number of function evaluations needed in an optimisation process. In this study, absolute search times are also used in the comparison.

The performances of the three learning evolutionary algorithms, a “Simple GA” (Goldberg, 1989) with tournament selection and generation gap maintenance, a conventional SA with single-point 100 thermo-equilibrium refinements (Michalewicz, 1994) and a Downhill Simplex method (Press *et al.*, 1994) are compared in this study. For each method, 10 experiments are carried out with randomly generated initial guess.

Reliability

The mean reliability of the various methods is shown in Figure 2.16, indicating that the three learning EAs are considerably more reliable than the GA, the SA and the Simplex method.

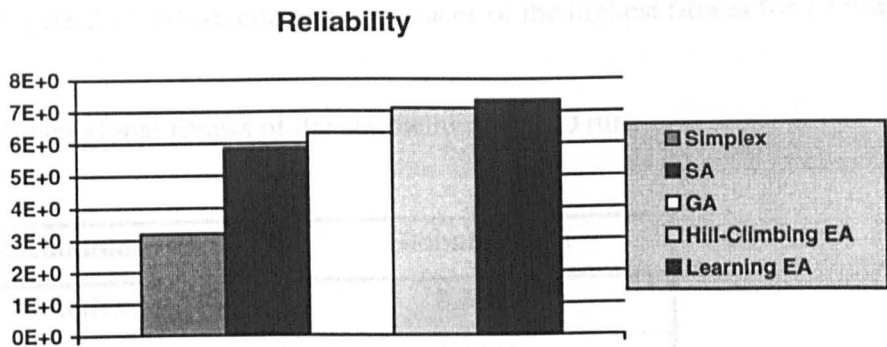


Figure 2.16 Mean reliability of the six methods for 10 runs



Convergence

Mean values of the highest fitness are shown in Figure 2.17 against the absolute time. It can be seen that the gradual learning EA converges more accurately and slightly faster than the GA, with the hillclimbing and learning EA catching up quickly. Note that a single run gives a very close convergence trace to the averaged one. As also shown in Table 2.1, the three learning EAs have offered an accuracy unmatched by the simple GA, SA and Simplex method. Note that the global fitness of Simplex method in Table 2.1 is obtained for the function evaluations shown in Figure 2.18.

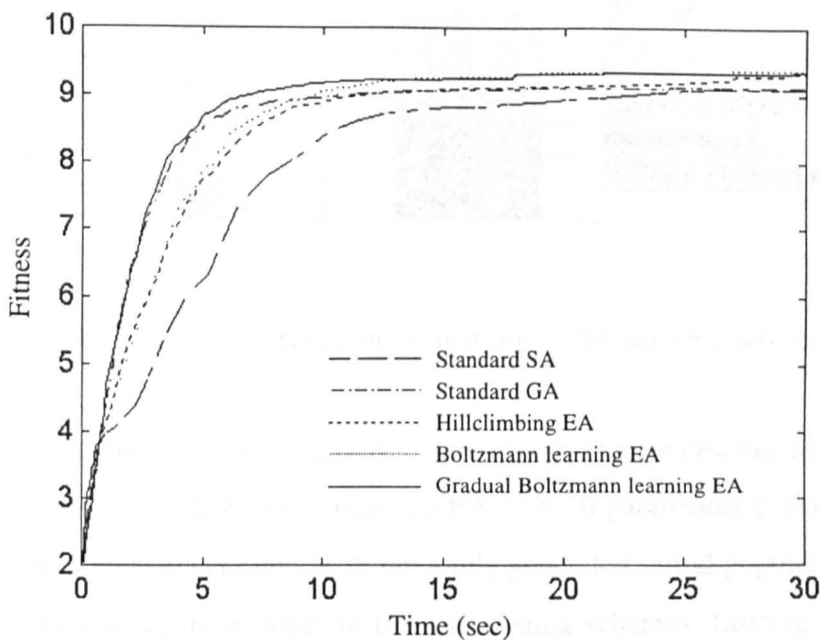


Figure 2.17 Mean convergence traces of the highest fitness for 10 runs

Table 2.1 Mean global fitness of the six methods for 10 runs

Methodology	Global fitness
Simplex	8.3059
Standard SA	9.1189
Standard GA	9.1297
EA with Hillclimbing	9.3154
Proposed learning EA	9.3360
Gradual learning EA	9.3204

Number of Function Evaluations

Figure 2.18 shows the mean number of function evaluations undertaken by the three learning EAs, simple GA and SA in 30 seconds. Note that mean number of function evaluations for the Simplex method was measured in 10 seconds. Compared with the methods reported by Renders and Bersini (1994), the learning EAs presented in this thesis take a similar number of function evaluations to produce more accurate results.

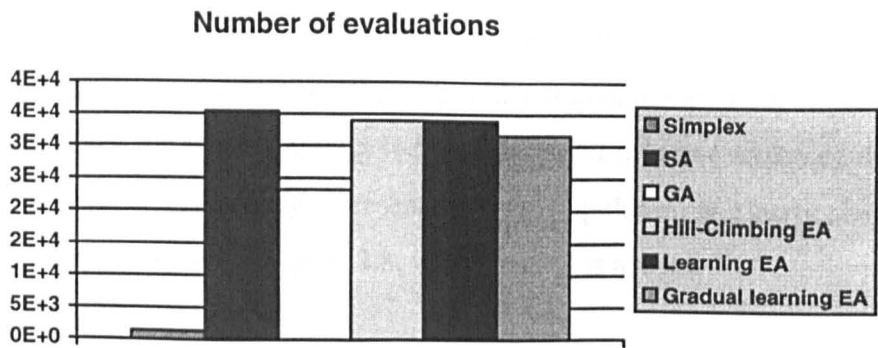


Figure 2.18 Mean number of function evaluations of the six methods for 10 runs

This benchmark test has also been employed to compare the processing time of various coding schemes reported in this thesis. Here, an EA with 10 parameters coded in the range of  $[-\pi , \pi]$  was run for 50 generations with randomly generated initial population. This was repeated three times using three different types of coding schemes, floating-point coding, decimal (integer) coding and one-integer-one-parameter coding, respectively. A comparison of processing times for the three different coding schemes to complete the 50 evolution is shown in Table 2.2. Apart from the reduction in memory usage, it is obvious that the proposed one-integer-one-parameter coding scheme has the least computational time, which only requires half of the processing time needed by a decimal (integer) coding scheme, and one quarter of the processing time required by a floating-point coding scheme.

Table 2.2 A processing time comparison of various coding schemes

Coding Schemes	Processing time (seconds)
Floating-point	112.96
Decimal (Integer)	43.88
One-integer-one-parameter	26.87

2.9 Multi-Objective Evolutionary Algorithms

2.9.1 A Tournament Niche MOEA

This thesis has implemented a MOEA with tournament selection based on the Pareto optimality cost-assignment scheme proposed by Fonseca and Fleming (1993). This approach (Tan and Li, 1997b) avoids the need of sorting, interpolating and fitness averaging in every generation as required by the algorithms from Fonseca and Fleming (1993) and thus can relatively save the required computational time. Since tournament selection determines the winner among individuals based on comparison of their costs or fitness, not all ranks are necessary represented in the population at a particular generation. An example of this is shown in Figure 2.8, where rank 3 is absent.

In order to spread the individuals equally along the Pareto surface, the ‘niche induction’ technique is used and is implemented by a sharing function as described in Section 2.5.1. Mating restriction discussed in Section 2.5.2 is applied here, as arbitrary mating of individuals may decrease the non-dominated individuals along the Pareto surface. Variation of the niching Pareto MOEA from a simple EA is shown in Figure 2.19. Deterministic tournament selection is used in which individuals are divided into subgroups and the individuals with the smallest cost among the subgroup is selected. Unlike other EA toolboxes for Matlab, this MOEA can be easily compiled into stand-alone C++ source codes using Matcom (MathTools, 1997) to significantly reduce the execution time. Here, a simple graphical user interface (GUI) as shown in Figure 2.20 has been written in M-files to make use of the Matlab graphical capabilities (MathWorks, 1992).

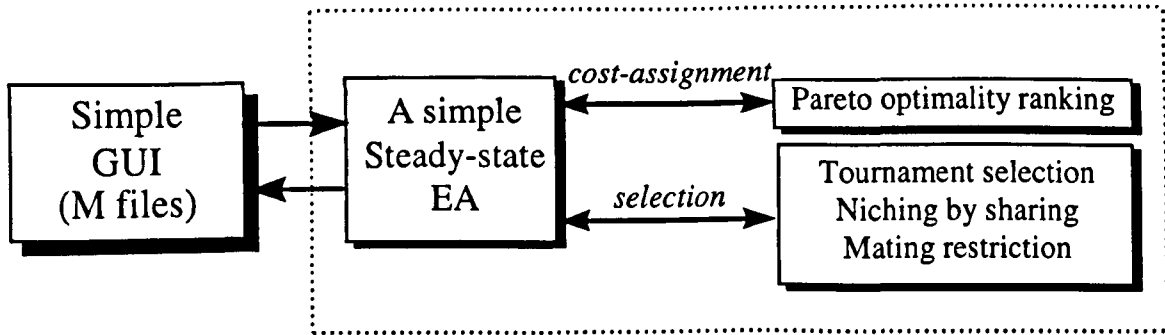


Figure 2.19 A niche Pareto tournament selection MOEA

Type of MOEA		Coding	
<input checked="" type="radio"/> Steady State		<input checked="" type="radio"/> Decimal	
<input type="radio"/> Regular		<input type="radio"/> Integer	
Obj. Function	objfunc.m		
Mins Vector	2		
Maxs Vector	2		
Goal Initial	00		
Goal Priority	11		
# of Objectives	2		
# of Parameters	8		
SIMULINK SETUP			
ULTIC		Modelling	
Need Some Help ?			

# of Generations	100
Population Size	100
Prob. of Mutation	0.01
Prob. of Crossover	0.7
Crossover Points	2
Niching (No/Yes)	Yes
Assumed Peaks	2
Selection	Tournament
Tournament Size	4
SS Population Size	20
Mating Restriction	Yes
Migration (No/Yes)	Yes
# of Migration	10
FineTuning	Sim-Annealing

<< Back
Quit MOEA ?
Accept + Run MOEA >>

Figure 2.20 A tabular graphical user interface for the MOEA

### 2.9.2 Optimisation by Tournament Niche MOEA

To study the performance of the MOEA developed in this thesis, consider the minimisation of the following two objectives (Fonseca and Fleming, 1995):

$$f_1(x_1, \dots, x_n) = 1 - \exp\left(-\sum_{i=1}^n \left(x_i - \frac{1}{\sqrt{n}}\right)^2\right) \quad (2.17)$$

$$f_2(x_1, \dots, x_n) = 1 - \exp\left(-\sum_{i=1}^n \left(x_i + \frac{1}{\sqrt{n}}\right)^2\right)$$

Due to the symmetry of the two functions  $f_1$  and  $f_2$ , the Pareto-optimal set are the points on the line defined by

$$x_1 = x_2 = \dots = x_n \wedge \frac{-1}{\sqrt{n}} \leq x_1 \leq \frac{1}{\sqrt{n}} \quad (2.18)$$

The number of parameters  $n$  is set to 8 in this experiment. The parameters are decimal coded in the interval  $[-2,2)$  and concatenated to form the chromosomes. Figure 2.21 shows a trade-off surface corresponding to Equation (2.17).

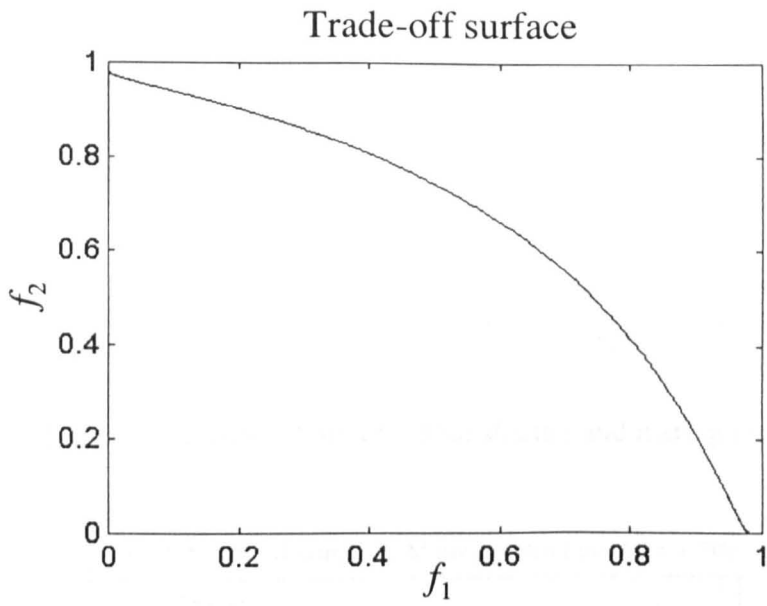


Figure 2.21 Trade-off surface of the multi-objective optimisation

To show how sharing and mating restriction together can significantly contribute to the performance of the EA, the MOEA with a population size of 100 was run for 100 generations without sharing or mating restriction. Multi-objective cost-assignment was performed as described and illustrated earlier in Figure 2.8. It can be seen that the population tends to concentrate on a small region of the trade-off surface as shown in Figure 2.22. If however, sharing and mating restriction are implemented as shown in Figure 2.23, the population is able to remain distributed and evolve across the entire trade-off surface after running the MOEA for 50 generations. Goal information can also be accommodated to drive the population to sample a given region of the trade-off surface. As shown in Figure 2.24, the MOEA was run for another 50 generations with a goal setting of  $\{g_1, g_2\} = \{0.6, 0.8\}$ . Obviously, the population is seen to concentrate on the preferred region of the trade-off surface, as desired.

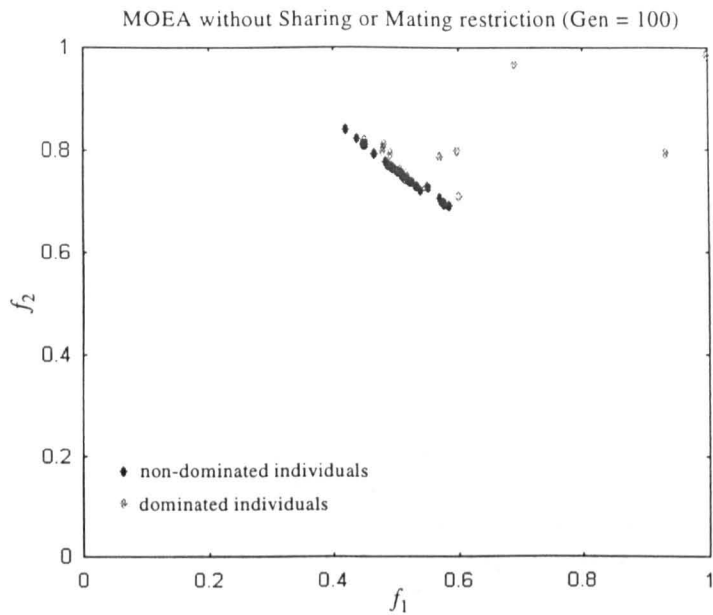


Figure 2.22 Multi-objective MOEA without sharing and mating restriction

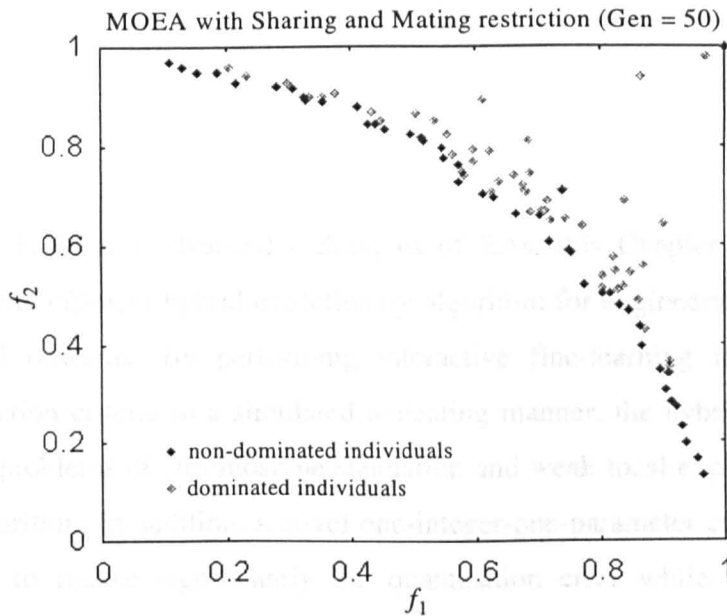


Figure 2.23 Multi-objective MOEA with sharing and mating restriction

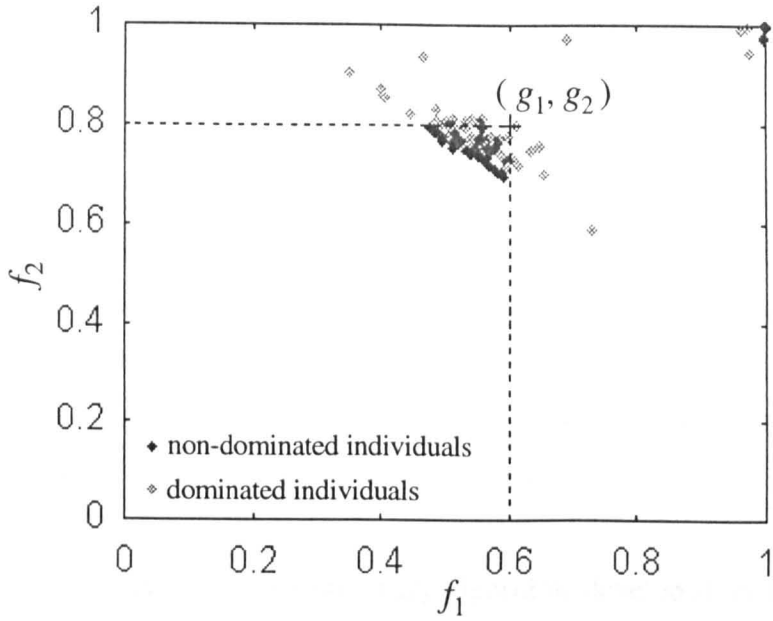


Figure 2.24 Zooming into a region of the trade-off surface by setting goals

$$\{g_1, g_2\} = \{0.6, 0.8\}$$

### 2.10 Summary

After presenting basic and advanced techniques of EAs, this Chapter has developed a highly accurate and efficient hybrid evolutionary algorithm for engineering applications in Section 2.7 and onwards. By performing interactive fine-learning incorporated with Boltmann’s selection criteria in a simulated annealing manner, the hybrid EA overcomes the well-known problems of chromosome stagnation and weak local exploration of a pure evolutionary algorithm. In addition, a novel one-integer-one-parameter coding scheme has been developed to reduce significantly the quantisation error while having a shorter chromosome length and better processing speed. An improved “Elitist Direct Inheritance” technique that utilises the “Choice Theorem” and reduces the *control parameters* in EAs has also been developed.

A multi-objective evolutionary algorithm has been studied. A tournament selection based on Pareto’s optimality cost-assignment scheme from Fonseca and Fleming (1993) has been proposed. This approach avoids the need of chromosome sorting, interpolating and fitness averaging in every generation as required by the algorithms from Fonseca and Fleming

(1993) and thus saves the overhead computational time. By incorporating a priority vector, the MOEA allows different preferences of the objective components and makes the optimisation transparent and visible. Niching and mating restriction techniques have also been incorporated in the MOEA for better evolution along the Pareto optimal surface. Parallelism of the developed hybrid EA has been studied. The parallel EA is based on an island model where multiple independent sub-populations each run a steady-state EA on their own and occasionally fitter chromosomes migrate between the sub-populations. This island model parallel EA was implemented on a network of transputers of a Parsytec SuperCluster, running parallel C in the PARIX environment.

A module hierarchy of the hybrid evolutionary algorithm developed in this Chapter is shown in Figure 2.25. Recommended routes of the hybrid EA are shown with solid links, while alternative routes are indicated by dotted links. This hybrid EA has been tested using a well-known benchmark problem and has been found to be highly accurate, effective, efficient and reliable. It is thus applied to solve all control engineering problems detailed in the following Chapters, where intelligent model fitting and computer-automated control system design are elaborated.



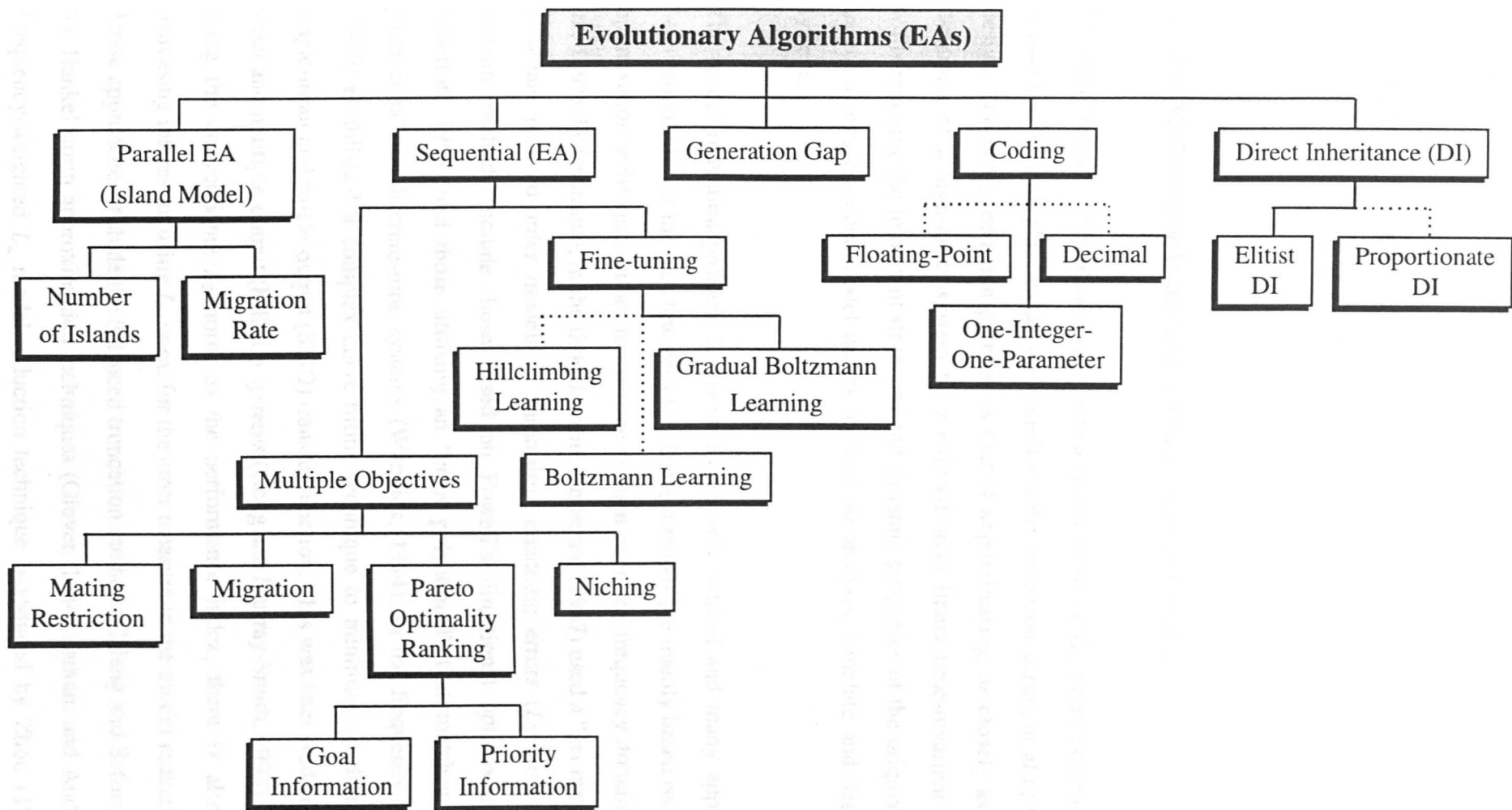


Figure 2.25 Module hierarchy of the hybrid EA for control engineering problems

## Chapter 3

# HIGHLY ACCURATE MODEL REDUCTION AND LINEARISATION BY LEARNING AND EVOLUTION

### 3.1 The Model Order Reduction and Linearisation Problem

For many modern control schemes, such as  $\mu$ -synthesis or  $H_\infty$  based control, it is usually required to perform plant model or controller order reduction during or after the process of design. The order reduction problem is that of approximating, as closely as possible, the dynamics of a high-order system by a reduced-order linear time-invariant (LTI) model, while retaining the important structural and dynamic properties of the original system. The use of a reduced-order model makes it easier to analysis, simulate and design a control system.

The model reduction problem has been extensively studied and many approaches have been proposed over the past few decades. These methods are mainly based on conventional optimisation techniques either in the time domain or in the frequency domain alone (Li *et al.*, 1997). For example, in the time domain, Anderson (1967) used a “geometric approach” to obtain reduced-order models of minimised quadratic errors ( $L_2$  norm). Other time domain methods include those based on Powell’s directional optimisation (Xue and Atherton, 1993) and those utilising an “error polynomial” that matches the “Markov parameters” for discrete-time systems (Warwick, 1984). In the frequency domain, Levy (1959) established a complex curve-fitting technique to minimise quadratic errors of a single-input and single-output (SISO) transfer function. This was later studied for multiple-input and multiple-output (MIMO) systems (Gong and Murray-Smith, 1993). In addition to using the conventional  $L_2$  norm as the performance index, there is also recently an increasing interest in using  $L_\infty$  norm for the error measure in the model reduction problems. These approaches include the balanced truncation method (Chiang and Safonov, 1992) and the Hankel norm approximation techniques (Glover, 1984; Latham and Anderson, 1986). Frequency-weighted  $L_\infty$  model reduction technique introduced by Zhou (1995) offers a

good  $L_\infty$  norm error bounds, but the resulting reduced model may still only be local solution.

These reduction methods are mainly based on gradient-guided type parameter estimation that are essentially guided by *a-priori* directions on the error surface of the cost function and are thus noise-prone (Li *et al.*, 1997). For this, the cost function need to be differentiable or well-behaved and the order of the reduced model be known *a-priori*. Although such *directed* optimisation techniques still serve as the major optimisation tool in control systems engineering, following an *a-priori* direction may not lead to *globally* optimised model parameters (Renders and Bersini, 1994), as the unknown error surface is usually multidimensional. To achieve good reduction tractability and quality, optimisation methods that rarely rely on a “wise” selection of initial conditions or *a-priori* parameters of the reduced model are needed.

A similar problem to the model reduction is the linearisation of physical nonlinear plants with differential equations structured by nonlinear dominant dynamics. This linearisation process is equivalent to reduce the “order” of a nonlinear system, which is in effect an infinite-order ‘linear’ system. The difference between reduction from an infinite-order and that from a finite-order lies in the need of a set operating point of the nonlinear system. Linearisation of a nonlinear system is often carried out using a calculus based Taylor expansion around an equilibrium operating point. Such a method is only valid if the nonlinearities of the plant are differentiable in engineering practice (Franklin *et al.*, 1991). The linearisation process becomes more difficult when the plant is a MIMO system. Further, this linearisation method may only be suitable if the plant operates in a relatively small or linear region.

The hybrid evolutionary method developed in Chapter 2 is applied here to solve the model order reduction and linearisation problems. Compared with Powell’s technique, it has been shown that EAs offer better accuracy and reliability in multidimensional optimisations (Renders and Bersini, 1994). The evolutionary methods, in their various forms, have successfully been applied to problems of model order reduction (Caponetto *et al.*, 1994; Li *et al.*, 1997; Yang *et al.*, 1996) and linearisation of nonlinear plants (Tan *et al.*, 1995, 1996).

Section 3.2 develops a  $L_2$  norm model reduction technique for systems and control engineering applications based on the Boltzmann learning refined evolution method of Chapter 2. This method is applicable to generic applications in both the discrete and the continuous-time SISO and MIMO systems. Enabled by a control gene for structure switch, the technique is also for simultaneously recommending both an optimal order number and optimal parameters. Evolution based techniques for multivariable  $L_2$  system linearisation from plant input-output behaviour in both the time and the frequency domain are developed in Section 3.3. Comparison of the hybrid EA linearisation and classical Taylor expansion linearisation approach is studied. The method is extended to inverse model linearisation and linear approximate-model network which allows linearisation for an entire operating region of the nonlinear system. Section 3.4 presents an evolutionary  $L_\infty$  model reduction method that offers a tighter  $L_\infty$  error bound and applicable to both continuous and discrete-time systems. Benchmark example shows that this evolutionary  $L_\infty$  model reduction method provides a superior performance than existing methods studied in this thesis. Evolutionary  $L_\infty$  linearisation techniques in the frequency domain is studied in Section 3.5. A summary is given in Section 3.6.

## **3.2 $L_2$ Model Reduction by Evolution**

### **3.2.1 Reduction Based on a Time Domain Cost**

Consider an  $m^{\text{th}}$  order discrete-time system described in Z-transform by

$$G(z) = \frac{b_n z^n + b_{n-1} z^{n-1} + \dots + b_0}{z^m + a_{m-1} z^{m-1} + \dots + a_0} \quad (3.1)$$

where  $n < m$  for a causal discrete-time system. Suppose a reduced-order model approximating this system is represented by

$$G_\theta(z) = \frac{d_k z^k + d_{k-1} z^{k-1} + \dots + d_0}{c_l z^l + c_{l-1} z^{l-1} + \dots + c_0} \quad (3.2)$$

where

$$0 < k < l \leq m-1 \quad (3.3)$$

Alternatively, the numerators and denominators of Equations (3.1) and (3.2) may be expressed in factorised forms. Note that the coefficients of the most significant order terms,  $c_l z^l$  and  $d_k z^k$ , are not pre-normalised to 1, so that these coefficients are allowed to be zero in an optimal reduction. Note also that, however, coefficients of high-order terms may not automatically become zero in conventional optimisations, since relatively higher-order models tend to offer relatively lower errors and conventional techniques are poor in approximating zero parameters.

To formulate the reduction problem, first define the parametric vector corresponding to  $G_\theta$  as

$$\theta = [c_0, \dots, c_l, d_0, \dots, d_k]^T \quad (3.4)$$

Step inputs that often used for model reductions in the time domain is adopted here. Suppose both the original and the reduced models are excited by the identical unit step signal  $u(t)$ . The discrepancy between their responses is given by

$$h(t) = [G(z) - G_\theta(z)] u(t) \quad (3.5)$$

This discrepancy is usually used to assess the closeness of the two models (Warwick, 1984; Xue and Atherton, 1993). Here the Z-transform argument  $z$  is also used to represent the unit forward-shift operator in the time domain. The reduced model may thus be obtained by minimising the following cost function (Warwick, 1984)

$$J(\theta) = \sum_{t=1}^N [w(t)h(t)]^2 \quad (3.6)$$

where  $w(t)$  is a weighting function with respect to time. Here the Euclidean distance in the  $l_2$  space is measured from  $t = 1$ , for a causal system, to  $t = N$ , the window size stretching to the steady-state. If lower order models are preferred and the cost of the order needs to be treated in the same degree as the quadratic errors, for example, Equation (3.6) may take the form of

$$J(\theta) = l \sum_{t=1}^N [w(t)h(t)]^2 \quad (3.7)$$

Here the range of  $k$  and  $l$  are limited by Equation (3.3). If the order number needs to be treated distinctively, it can form a second objective in addition to Equation (3.6). The cost formulations can also be extended to represent continuous-time and multivariable systems. Although quadratic errors or the  $L_2$  norms are most widely used as a measure of fitting quality in model reduction and linearisation applications, other measures such as application-oriented norms or  $L_\infty$  norm can also be used. The use of  $L_\infty$  norm in the model reduction will be discussed later in Section 3.4.

### 3.2.2 A Time Domain Model Reduction Example

Consider the fourth-order discrete-time system studied by Warwick (1984) and later by Xue and Atherton (1993) as follows:

$$G(z) = \frac{0.3124z^3 - 0.5743z^2 + 0.3879z - 0.0889}{z^4 - 3.233z^3 + 3.9869z^2 - 2.2209z + 0.4723} \quad (3.8)$$

For this benchmark problem, a second-order causal model is required to approximate the discrete-time model given in Equation (3.8). Note that the Xue and Atherton (1993) method required a good initial starting point via transforming the discrete-time system to a continuous-time system before Powell's optimisation technique was employed.

The hybrid EA developed in Chapter 2 has been applied here to solve this fixed-order model reduction problem. The parameter sets in the initial population were generated

randomly, i.e., no *a-priori* knowledge was required. The parameters converged rapidly, as shown in Figure 3.1. The algorithm was coded in Turbo Pascal 7.0 and it took, on average, 2.5 minutes on a 50 MHz 80486 DX2 processor to complete 100 generations. The step responses of the original system and the reduced model are plotted in Figure 3.2, which indicates a good reduction performance. The RMS error and the parameters of the reduced system obtained by the hybrid EA are shown in Table 3.1. For comparison, the parameters of the reduced models and the resulting RMS errors obtained by Warwick (1984) and Xue and Atherton (1993) using existing optimisation tools are also shown in Table 3.1. It can be seen that the proposed method performs better than the ‘error polynomial’ approach of Warwick (1984). It also offers a slightly (1.6 %) improved fitting quality in terms of RMS errors than that obtained by Xue and Atherton (1993), without the need to transform the system to a continuous-time system and the need of an *a-priori* starting point.

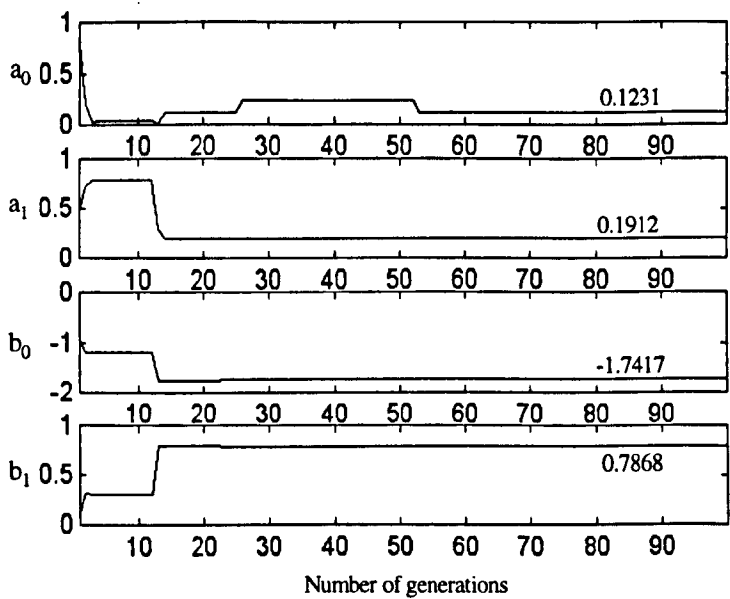


Figure 3.1 Convergence traces of the parameter set with the lowest fitting error

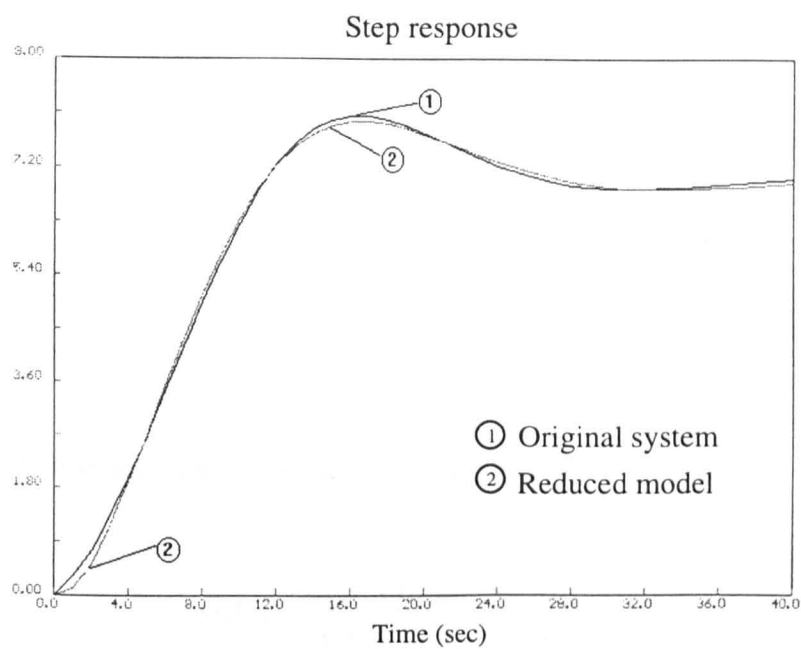


Figure 3.2 Step responses of the original and the evolved lower-order models

Table 3.1. The parameter sets and RMS errors obtained by various methods for Equation (3.8)

Parameters	Warwick	Xue and Atherton	Boltzmann Learning EA
$c_0$	0.3124	0.1299	0.1231
$c_1$	-0.0298	0.1820	0.1912
$d_0$	-1.7369	-1.7431	-1.7417
$d_1$	0.7773	0.7877	0.7868
$c_2 = 1, d_2 = 0$ (reducing to a causal second-order model is required here)			
RMS error	0.1884	0.0835	0.0822



### 3.2.3 Reduction Based on a Frequency Domain Cost

In the frequency domain, a transfer function  $G(s)$  corresponding to  $G(z)$  is used to represent the system, as given by

$$G(s) = \frac{b_n s^n + b_{n-1} s^{n-1} + \dots + b_0}{s^m + a_{m-1} s^{m-1} + \dots + a_0} \quad (3.9)$$

The transfer function representing a reduced model is

$$G_\theta(s) = \frac{d_k s^k + d_{k-1} s^{k-1} + \dots + d_0}{c_l s^l + c_{l-1} s^{l-1} + \dots + c_0} \quad (3.10)$$

where  $n < m$  for a causal system and  $k$  and  $l$  satisfy Equation (3.3). In the frequency domain, an error transfer function may be defined by

$$H(s) = G(s) - G_\theta(s) \quad (3.11)$$

For a SISO system, an order reduction problem is to find an optimal  $G_\theta(s)$  such that it minimises a cost function as given by:

$$J(\theta) = l \sum_{f=1}^N \left[ W(j\omega_f) H(j\omega_f) \right]^2 \quad (3.12)$$

Here  $W(j\omega)$  is not a transform of  $w(t)$ . It is a frequency weighting function which allows the fitting errors in the chosen parts of the frequency range to be emphasised if needed.

Similarly, for a MIMO system, a weighted cost summing up all squared Euclidean error norms

$$J(\theta) = \sum_p \sum_q J_{pq}(\theta) \quad (3.13)$$

can be used, where

$$J_{pq}(\theta) = l \sum_{f=1}^N \left[ \left| W_{pq}(j\omega_f) H_{pq}(j\omega_f) \right| \right]^2 \quad (3.14)$$

with  $W_{pq}(j\omega)$  being a weighting element. Here  $H_{pq}(j\omega)$  is an element of the error transfer function matrix, and  $G_{pq}(j\omega)$  terms in Equation (3.11) are usually pre-set to share an identical denominator. The metric window size needs to cover the relevant frequency range for the required reduction. Note that, instead of using the summation of Euclidean norms, the Frobenius or the  $H_2$  norm may also be used for MIMO systems.

### 3.2.4 Open-Order Model Reduction in the Frequency Domain

The 8th-order transfer functions of a 1-input and 2-output continuous-time model of a fighter aircraft studied by Bacon and Schmidt (1988) and later by Gong and Murray-Smith (1993) are given below:

$$\begin{bmatrix} \frac{q(s)}{\delta(s)} \\ \frac{n_{zcr}(s)}{\delta(s)} \end{bmatrix} = \Delta^{-1}(s) \begin{bmatrix} 5.26s(s+0.0103)(s+0.5)(s+1.887)(s+13.986) \\ 1.34s(s+0.00066)(s+49.99)(s+0.5)(s+1.887)(s+13.986) \end{bmatrix} \quad (3.15)$$

where

$$\Delta(s) = (s+0.418)(s+1.34)(s^2+0.00264s+0.006724)(s^2+3.3916s+7.7284) \cdot (s^2+33.0576s+290.3611) \quad (3.16)$$

This example represents the longitudinal responses in terms of pitch rate  $q(s)$  and the normal acceleration  $n_{zcr}(s)$  to elevator stick force  $\delta(s)$ . It is noted that the dynamics in phugoid and lateral spiral modes are covered by the frequency range  $[0.5, 15]$  rad s<sup>-1</sup> and thus this range has been used in model reduction studies (Bacon and Schmidt, 1988; Gong and Murray-Smith, 1993; Li *et al.*, 1997).

The hybrid EA detailed in Chapter 2 has been applied to the original model to evolve reduced models for 10 minutes on a 100 MHz Pentium PC. The results are shown in the middle row of Table 3.2. For comparison, the transfer functions and RMS errors of a reduced third order model obtained by Gong and Murray-Smith (1993) are shown in the first row of Table 3.2. It can be derived that the evolved model provides an improved reduction quality by 7.4 % with respect to Equation (3.13). The frequency responses of the original system, the reduced model by Gong and Murray-Smith (1993) and the reduced model by evolution and learning are compared in Figure 3.3. Clearly the hybrid EA performs better than the conventional optimiser. To further verify the EA performance, impulse responses for both the original system and the reduced 3<sup>rd</sup> order models are plotted in Figure 3.4 and Figure 3.5, respectively. Although the cost minimised was based on the magnitudes (and not phases) of error transfers, the phase discrepancy of the evolved model is also smaller than that of the conventionally reduced model in the relevant frequency range. Note that, if minimising phase discrepancy is required in an application, it can also be incorporated in the cost function.

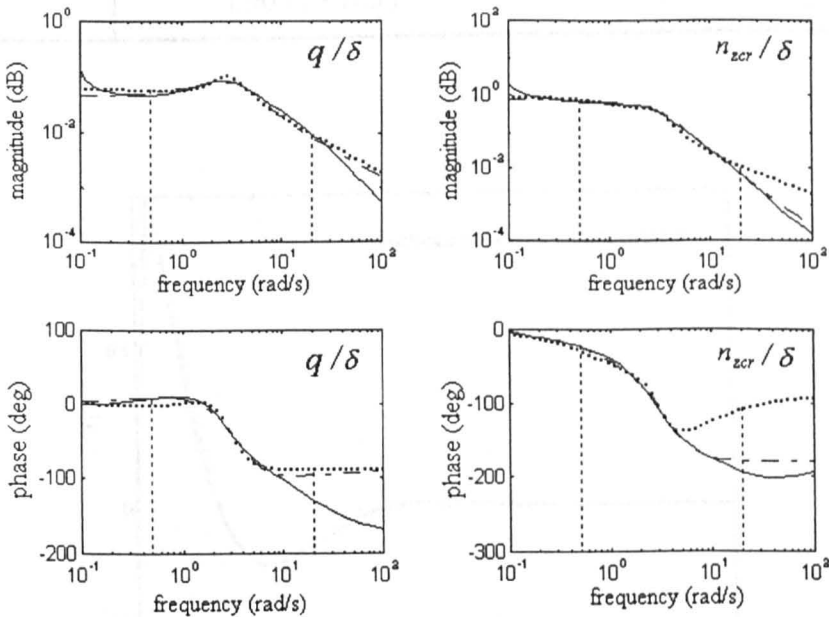


Figure 3.3 Magnitude and phase plots. Original  $G(s)$  { — }; Reduced order  $G_\theta(s)$  { .... } by Gong and Murray-Smith (1993); Reduced order  $G_\theta(s)$  { -.-. } by the hybrid EA

Table 3.2 Reduced-order transfer function and standard deviation matrices for the fighter aircraft of Equations (3.15) and (3.16)

Orders	Transfer function matrices	$[J_{\text{RMS}}(\theta)]$	$\sum_{pq} J_{\text{RMS}}^2(\theta)$	RMS of $H(j\omega)$
3 (Gong and Murray-Smith, 1993)	$\begin{bmatrix} \frac{0.18s^2 + 0.54s + 0.39}{s^3 + 2.96s^2 + 10.78s + 6.8} \\ \frac{0.2s^2 + 2.03s + 5.53}{s^3 + 2.96s^2 + 10.78s + 6.8} \end{bmatrix}$	$\begin{bmatrix} 0.0155 \\ 0.2427 \end{bmatrix}$	0.05914	$\begin{bmatrix} 32.5549 \\ 53.9169 \end{bmatrix}$
3 (fixed)	$\begin{bmatrix} \frac{0.15s^2 + 1.23s + 0.84}{s^3 + 5.7s^2 + 19.76s + 18.92} \\ \frac{2.49s + 13.61}{s^3 + 5.7s^2 + 19.76s + 18.92} \end{bmatrix}$	$\begin{bmatrix} 0.0144 \\ 0.2336 \end{bmatrix}$	0.05478	$\begin{bmatrix} 30.4882 \\ 10.0567 \end{bmatrix}$
1 (variable)	$\begin{bmatrix} \frac{5.77}{96.7s + 103} \\ \frac{79.4}{96.7s + 103} \end{bmatrix}$	$\begin{bmatrix} 0.0285 \\ 0.2355 \end{bmatrix}$	0.05627	$\begin{bmatrix} 39.5149 \\ 69.6141 \end{bmatrix}$

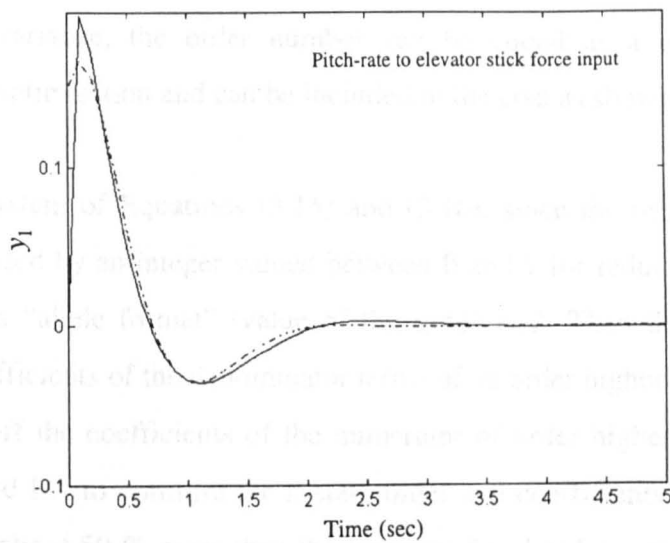


Figure 3.4 Impulse response of pitch-rate to elevator stick-force input

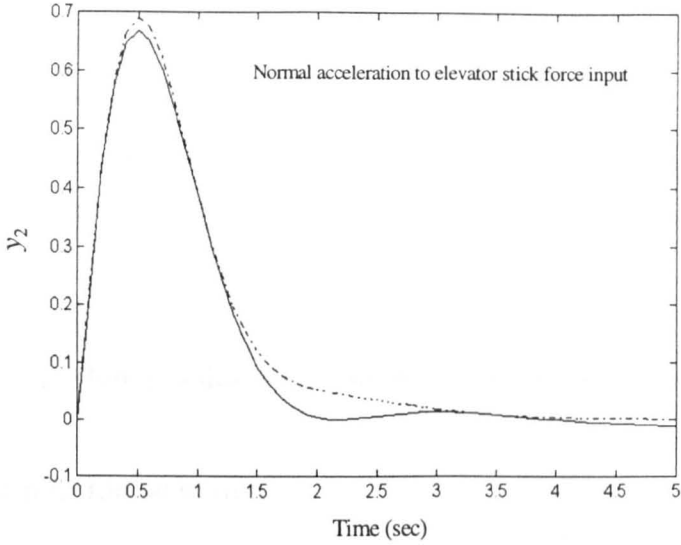


Figure 3.5 Impulse response of normal acceleration to elevator stick-force input

The above examples are based on fixed-order reductions. The coding version of an EA can, however, allow the order number to be encoded and thus optimised in the evolution process. This is done by a “*control gene*” that acts as a “structural switch”. It varies the number of the highest order of a candidate model and thus switches on and off the coefficients of the terms that have a higher order. Here the range of the order coded satisfies Equation (3.3). However, the EA realising this approach has often recommended the highest possible order, i.e.,  $l = m-1$ . This hardly achieves model reduction and is thus modified. For example, if the order number needs to be penalised in the same way as the total magnitude variance, the order number can be coded as a control gene, in the parameter set for optimisation and can be included in the cost as shown in Equation (3.14).

For the aircraft system of Equations (3.15) and (3.16), since the original order is 8, the control gene is coded by an integer valued between 0 and 7 for reduction. Suppose at one evolution stage its “allele format” (value of the gene) is 2. Then this control gene will switch off the coefficients of the denominator terms of an order higher than or equal to 3. It will also switch off the coefficients of the numerator of order higher than or equal to 2. Running the hybrid EA to optimise the system order and coefficients in the same process took 15 minutes (about 50 % more than the previous fixed-order reduction), although the number of parameters being optimised were doubled. For this variable-order reduction task, the evolution revealed that a first-order reduced model should be used due to penalty

on the order number. This model is shown in the last row of Table 3.2. It can be seen that it offers a reduction quality almost as high as the fixed third-order model, and better than that obtained by Gong and Murray-Smith (1993). It is not surprising to see that the EA did not recommend a second or higher-order model for the cost governed by Equation (3.14) because of the reasonable accuracy offered by the first-order model.

### 3.3 $L_2$ Linearisation: Model Reduction from an Infinite-Order

#### 3.3.1 Time Domain $L_2$ Linearisation

With the increase in computing power and interest in modern control systems, linear state space representation of nonlinear multivariable systems has gained great significance in modern design methods such as  $H_\infty$  control and  $\mu$ -synthesis. In this section, a nonlinear coupled liquid-tank system shown in Figure 3.6 is to be linearised in the time domain. Based on Bernoulli's equation of mass-balance, the dynamic structure of this multivariable system is dominated by the nonlinear framework given by:

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} -\operatorname{sgn}(h_1 - h_2) \frac{C_1 a_1}{A} \sqrt{2g|h_1 - h_2|} \\ \operatorname{sgn}(h_1 - h_2) \frac{C_1 a_1}{A} \sqrt{2g|h_1 - h_2|} - \frac{C_2 a_2}{A} \sqrt{2g|h_2 - H_3|} \end{bmatrix} + \begin{bmatrix} \frac{Q_1}{A} & 0 \\ 0 & \frac{Q_2}{A} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (3.17)$$

Deduction of the nonlinear model in Equation (3.17) can be found in Appendix A. Here the tanks are linked through a coupling pipe of an equivalent orifice area  $a_1$ ; the equivalent discharging area of Tank 2 is modelled by  $a_2$ ; the liquid level in Tank 1 is  $h_1$ ; that in Tank 2 is  $h_2$  with a physical constraint being  $h_2 > H_3$ , the equivalent height of both the coupling and discharging pipes;  $C_1$  and  $C_2$  are equivalent discharge constants;  $A = 100 \text{ cm}^2$  is the cross-sectional area of both tanks (which can be physically measured with a relatively high accuracy);  $Q_1$  and  $Q_2$  are the input flow rate per actuating volt of the power amplifiers for Tank 1 and Tank 2, respectively; and  $g = 981 \text{ cm s}^{-2}$  the gravitational constant.

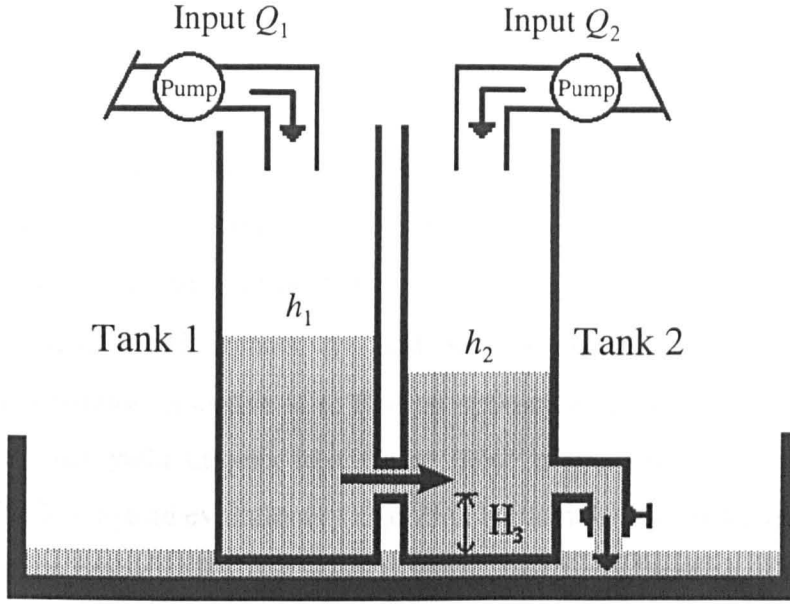


Figure 3.6 A nonlinear coupled liquid-level system.

To obtain a linear model of the nonlinear model of equation (3.17), linearisation by conventional approach around an equilibrium point using Taylor expansion was performed. Detailed derivation of this linear model is given in Appendix B. Based on manufacturer's specification and further physical measurements at equilibrium point of 0.185 m and 0.146 m of Tanks 1 and 2 respectively, a linear model is obtained,

$$\begin{bmatrix} \dot{h}_{11} \\ \dot{h}_{22} \end{bmatrix} = \begin{bmatrix} -0.05 & 0.05 \\ 0.05 & -0.101 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{22} \end{bmatrix} + \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

#### Linearisation by Step Excitation

##### A. Linearisation Around an Equilibrium Point with Small Perturbations

In this section, a second-order linear model is to be derived from the nonlinear model given by Equation (3.17). The linearised model is obtained from the first-order multivariable Taylor expansion as given by

$$\begin{bmatrix} \Delta \dot{h}_1 \\ \Delta \dot{h}_2 \end{bmatrix}_{H_1, H_2} = A \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \end{bmatrix} + B \begin{bmatrix} \Delta v_1 \\ \Delta v_2 \end{bmatrix} \quad (3.18a)$$

$$\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} + \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \end{bmatrix} \quad (3.18b)$$

where  $(H_1, H_2)$  is the equilibrium point around which linearisation is carried out. For this example, the equilibrium is obtained at the operating levels of 0.185 m and 0.146 m at the steady-state for Tanks 1 and 2, respectively. The output variation data,  $\Delta h_1$  and  $\Delta h_2$ , are collected when small PRBS signals,  $\Delta v_1$  and  $\Delta v_2$ , are applied to perturb the nonlinear system. The linearisation objective is to find an optimal set of state matrices  $A$  and  $B$  of Equation (3.18) that yield outputs best fitting those of the given nonlinear system of Equation (3.17). The hybrid evolutionary algorithm method detailed in Chapter 2 is applied to this linearisation problem. For this MIMO system, the fitness function used is

$$f = \frac{1}{1 + \{\|e_1\|_2 + \|e_2\|_2\}} \quad (3.19)$$

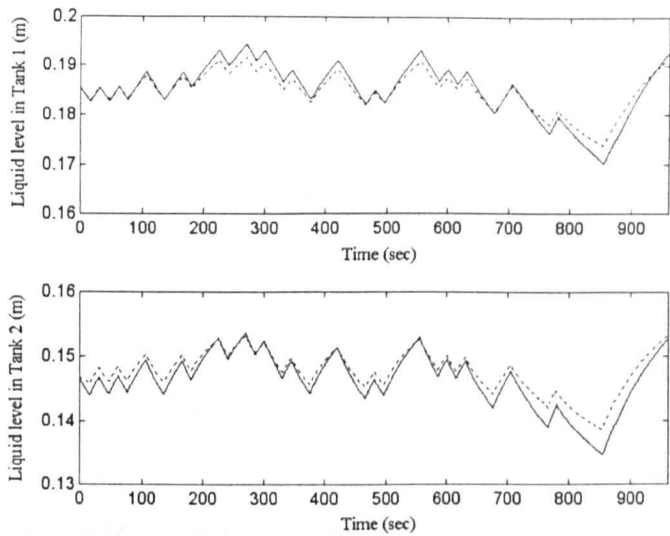
Here,  $e_1$  and  $e_2$  are errors between the actual and modelled liquid levels in Tanks 1 and 2, respectively. The evolved state matrices of the linearised model from the nonlinear model of equation (3.17) are given by:

$$A = \begin{bmatrix} -0.017 & 0.011 \\ 0.087 & -0.119 \end{bmatrix} \quad B = \begin{bmatrix} 85.49 & 0 \\ 0 & 85.49 \end{bmatrix}$$

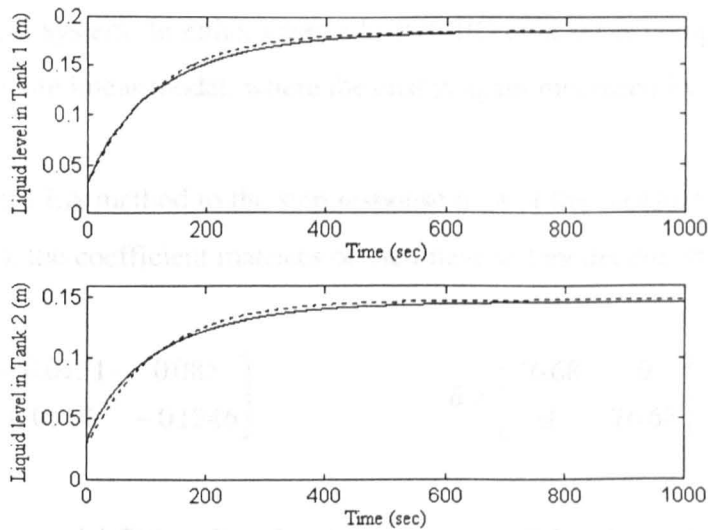
### Validation by Step Excitation

The responses of the linearised system and the original nonlinear system to the PRBS perturbations and to the steps that set the operating levels are depicted in Figure 3.7. It can be seen that the linearised model offers a reasonable fitting to the nonlinear system at the operating points. The resulting RMS error





(a) Responses to the PRBS perturbations



(b) Responses to the step inputs

Figure 3.7 Validation of the linearised system responses (dotted) against the original nonlinear system responses (solid)

Although this example shows linearisation in the state space representation, the method is equally straightforward to apply to linearising a nonlinear system to a transfer function matrix (TFM). The order of the TFM elements, as well as the structure of the system being identified, can also be coded in the chromosome without pre-specification and can be penalised in the performance index as described in Section 3.2.

## B. Linear Model Approximation

In some control engineering applications, a linear model may be required to approximate a nonlinear system (Tan *et al.*, 1995). Given an I/O data set from the nonlinear coupled liquid-level system, the linear model approximation is to find a linearised model

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix}_{H_1, H_2} = A \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + B \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (3.20)$$

such that it optimally approximates the nonlinear system. This means that a linearised model may be obtained either from the nonlinear model of Equation (3.17) or from the I/O data of the physical system. In either case, a known I/O data set is compared with the I/O data set of a candidate linear model, where the cost is again measured by Equation (3.19).

Applying the hybrid EA method to the step response data of the measured nonlinear model of Equation (3.17), the coefficient matrices of the linearised model are obtained as:

$$A = \begin{bmatrix} -0.0751 & 0.085 \\ 0.0914 & -0.1246 \end{bmatrix} \quad B = \begin{bmatrix} 76.68 & 0 \\ 0 & 76.68 \end{bmatrix}$$

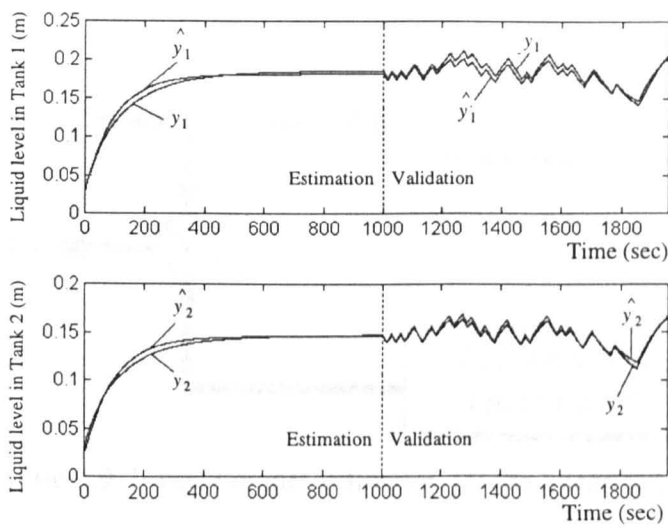
To carry out linear model fitting directly, step responses of the physical system have been sampled. These actual responses are then used to build a linear model governed by Equation (3.20). The coefficient matrices of the hybrid EA evolved linear model are:

$$A = \begin{bmatrix} -0.0683 & 0.0738 \\ 0.0708 & -0.0931 \end{bmatrix} \quad B = \begin{bmatrix} 63.06 & 0 \\ 0 & 63.06 \end{bmatrix}$$

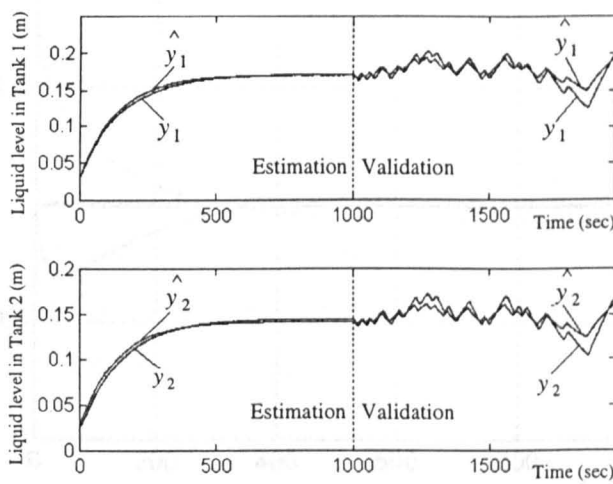
## Validation by Small Perturbations Around the Operating Points

For a validation around the set point, the steady-state operating point has been perturbed by small PRBS signals additive to the step excitation. The responses of the nonlinear model and of the physical system to such inputs are shown by the curves represented by  $y_1$  and  $y_2$

in Figure 3.8 and the outputs of the linearised models by  $(\hat{y}_1, \hat{y}_2)$ . It can be seen that both identified models offer a reasonable fitting to the nonlinear system.



(a) Validation of the linearised model evolved from the nonlinear model



(b) Validation of the linearised model evolved from the physical system

Figure 3.8 Outputs of a linearised model  $(\hat{y}_1, \hat{y}_2)$  validated against known data  $(y_1, y_2)$

**Validation by Inverse Model Simulation**

Inverse model validation technique (Thomson and Bradley, 1988) has been applied here to further validate the obtained linearised model. This validation approach is depicted in Figure 3.9, in which the nonlinear output data are used for the inverse simulation of the

linearised model. The resulting inverse model inputs are then compared with the actual applied nonlinear system inputs for validation. Figure 3.10 shows the inverse model input responses of  $v_1$  and  $v_2$ . It can be seen that both inverse model input responses are reasonable close to the actual applied step inputs.

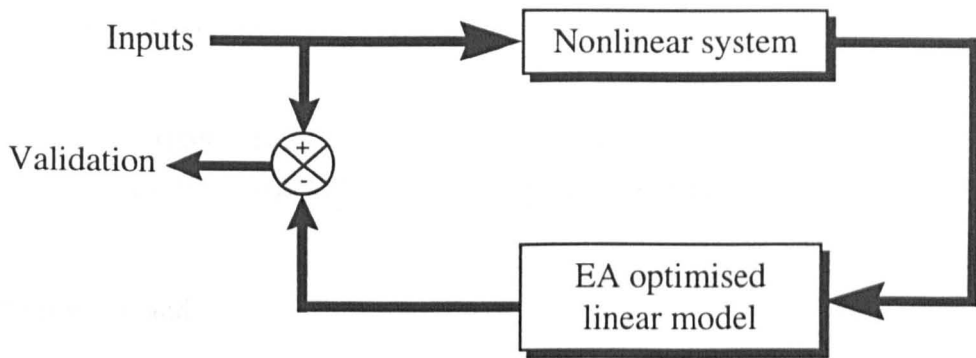


Figure 3.9 Inverse model validation for the linearised model

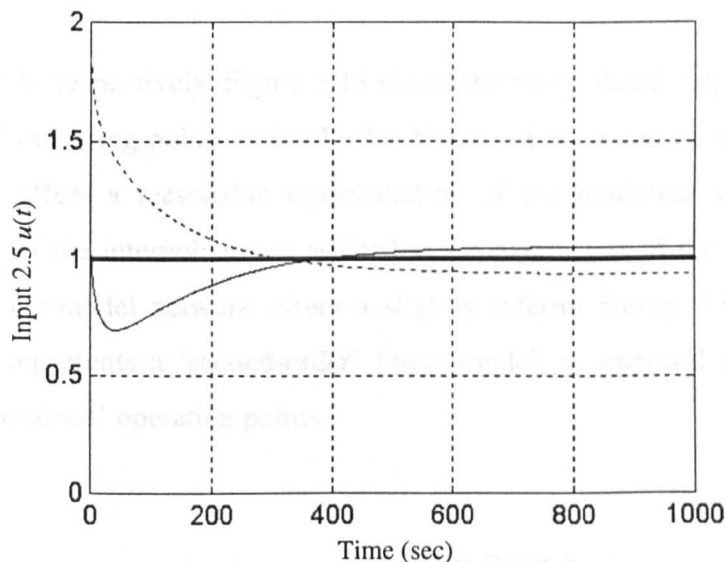


Figure 3.10 Input responses of the inverse model validation  $v_1(t)$  { — };  $v_2(t)$  { - - - }

C. Linear Approximate-Model Network

In this section, linear approximate-model network as shown in Figure 3.11 is studied. This approach is similar to the local model network, which allows the linearisation to be applied to more than one important operating point of a nonlinear system (Gawthrop, 1995; Gollee

*et al.*, 1994; Johansen and Foss, 1992, 1993; Murray-Smith, 1994). By interpolation of the parameters or outputs of the linear models, a network of linear approximate models can be formed easily as illustrated in Figure 3.12. To form a linear approximate-model network of the nonlinear system of Equation (3.17), two linear models for operating points  $v_{11} = v_{12} = 2 \text{ u}(t) \text{ V}$  and  $v_{21} = v_{22} = 3 \text{ u}(t) \text{ V}$  are evolved. The evolved coefficient matrices of the individual linear models are:

$$A = \begin{bmatrix} -0.065 & 0.069 \\ 0.028 & -0.047 \end{bmatrix} \quad B = \begin{bmatrix} 81.58 & 0 \\ 0 & 81.58 \end{bmatrix}$$

for  $v_{11} = v_{12} = 2 \text{ V}$  and

$$A = \begin{bmatrix} -0.013 & 0.008 \\ 0.049 & -0.072 \end{bmatrix} \quad B = \begin{bmatrix} 82.18 & 0 \\ 0 & 82.18 \end{bmatrix}$$

for  $v_{21} = v_{22} = 3 \text{ V}$ , respectively. Figure 3.13 shows the interpolated outputs of the network for the ‘unseen’ operating points at 2.2 V, 2.5 V and 2.8 V. It can be seen that the linear model network offers a reasonable representation of the nonlinear system for a wide operating range. If the interpolation is applied to the parameters of the linear models, the linear approximate-model network offers a slightly inferior fitting. This is because the ‘network’ now represents a ‘second-order’ linear model, as opposed to a ‘fourth-order’ network, for ‘untrained’ operating points.

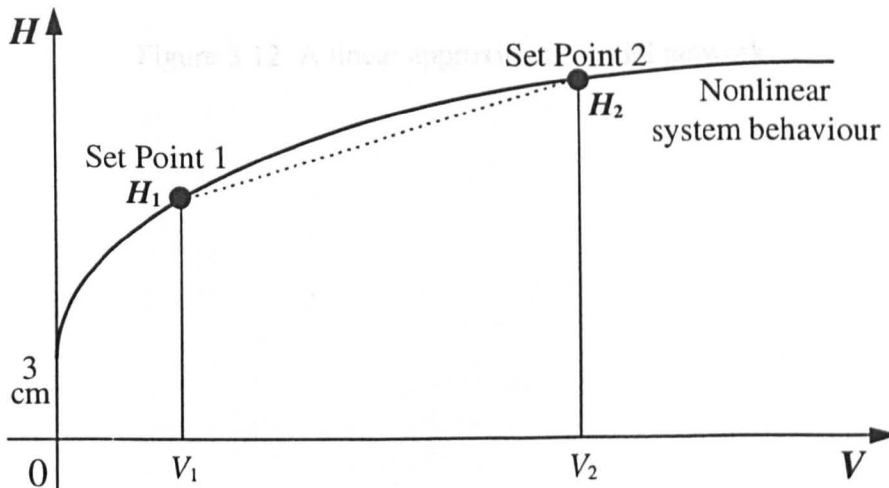
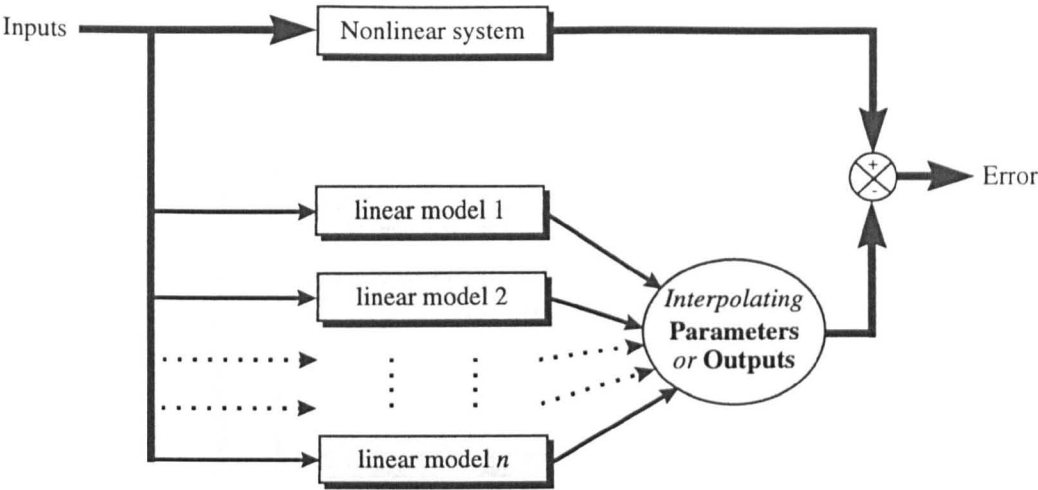
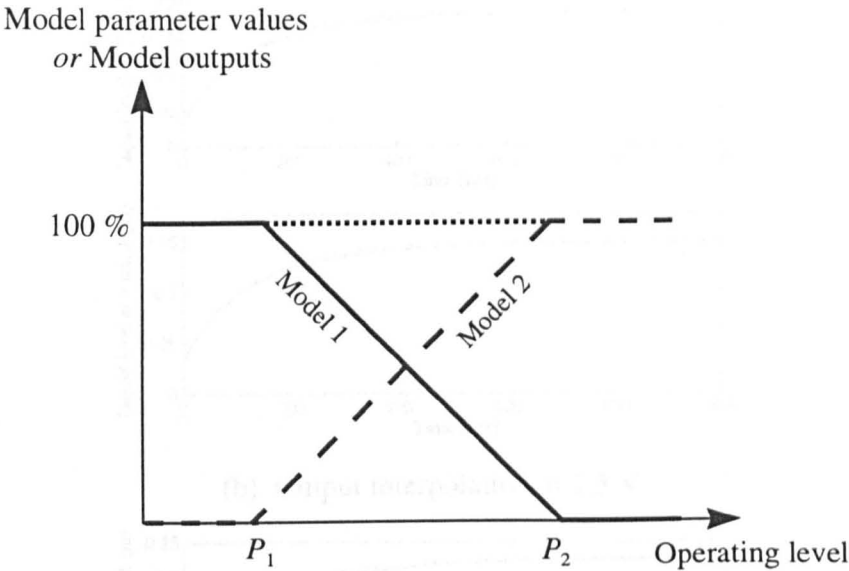


Figure 3.11 A linear approximate-model network

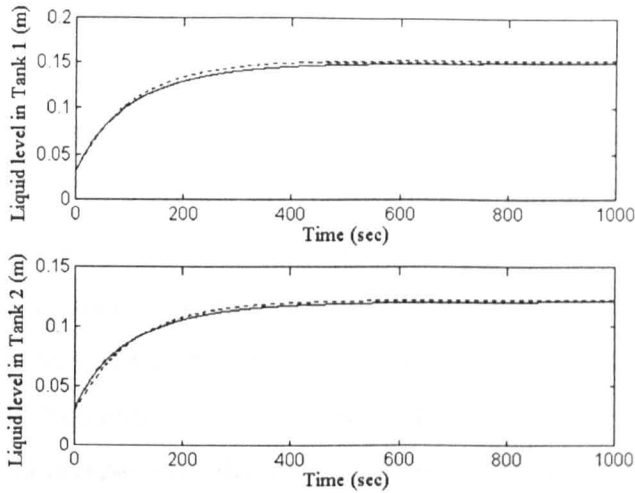


(a) A linear approximate-model network

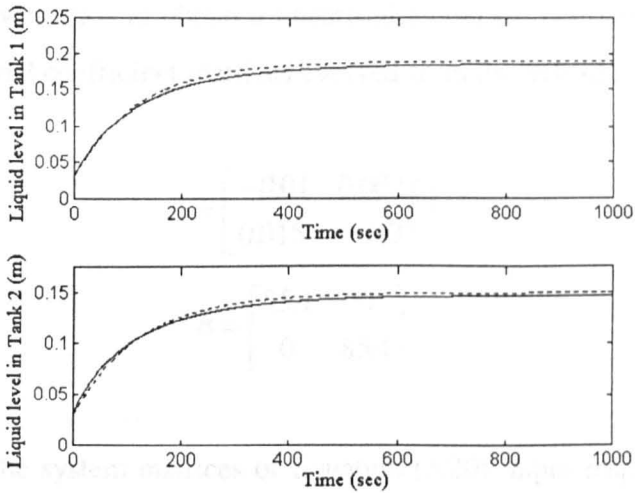


(b) Interpolation of models

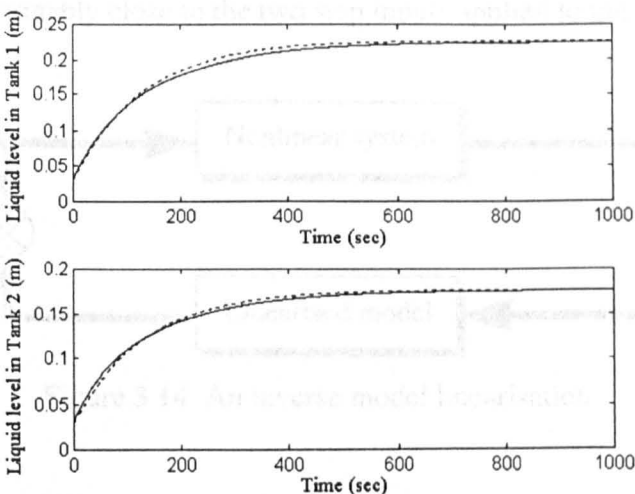
Figure 3.12 A linear approximate-model network



(a) Output interpolation at 2.2 V



(b) Output interpolation at 2.5 V



(c) Output interpolation at 2.8 V

Figure 3.13 Validation of linearisation by a linear approximate-model network of linear models at 2 V and 3 V

### D. Inverse Model Linearisation

In this Section, the inverse model simulation technique is applied to the linearisation problem. Similar to Section 3.3.1(B), the linearisation is carried out along an operating trajectory. Given a set of output response data of a nonlinear system, the purpose of inverse model linearisation is to obtain a linearised model by going through an inverse simulation to find a candidate model that best matches the input data, as illustrated in Figure 3.14. The cost function given by Equation (3.19) is adopted. However, here  $e_1$  and  $e_2$  are the errors between the actual and inverse modelled input voltages to Tanks 1 and 2, respectively. To illustrate the methodology, the same I/O data set of the nonlinear model of Equation (3.17) in Section 3.3.1(B) was used to obtain a linearised model for operating points  $v_{11} = v_{12} = 2.5 \text{ u}(t) \text{ V}$ . The optimal coefficient matrices evolved from the hybrid EA are

$$A = \begin{bmatrix} -0.01 & 0.0022 \\ 0.015 & -0.03 \end{bmatrix} \quad (3.21a)$$

$$B = \begin{bmatrix} 85.4 & 0 \\ 0 & 85.4 \end{bmatrix} \quad (3.21b)$$

where  $A$  and  $B$  are the system matrices of Equation (3.20). Input responses of the inverse simulation for the obtained linearised model are shown in Figure 3.15. It can be seen that the responses are reasonably close to the two step inputs applied to the nonlinear system.

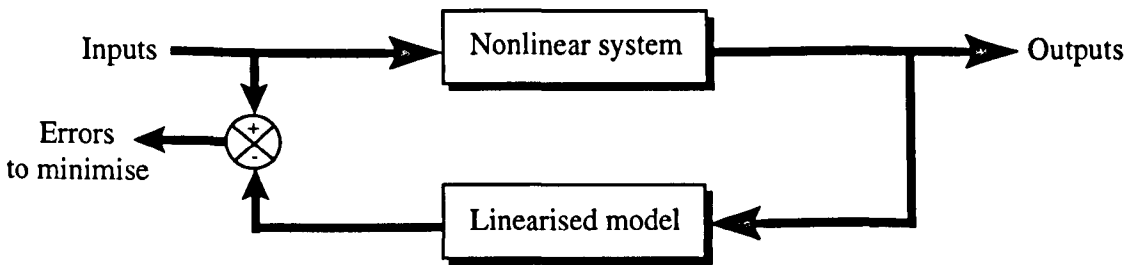


Figure 3.14 An inverse model linearisation



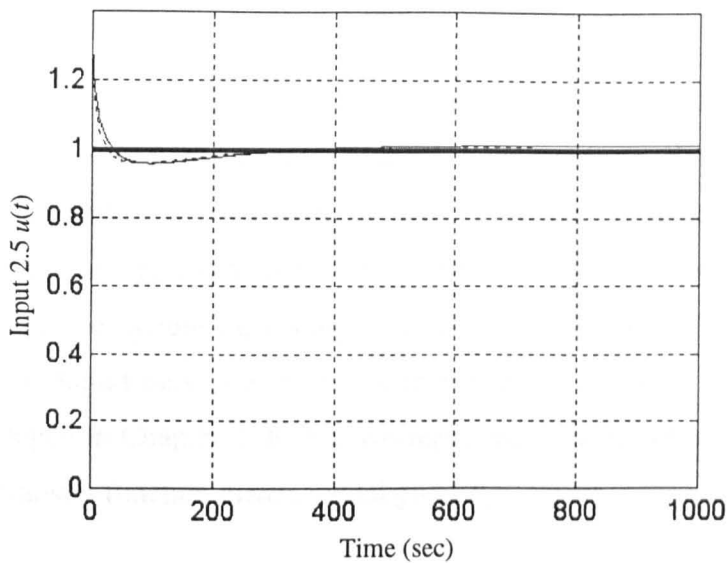


Figure 3.15 Input responses of the inverse simulation for the linearised model.  $v_1(t)$  {—};  $v_2(t)$  { - - - }

**Validation by Forward Simulation**

To validate the linearised model given by Equations 3.21 (a) and (b), forward simulation of this linear model has been carried out and the output responses are shown in Figure 3.16. The result that validates the linearised model is commendable, with a good representation of the nonlinear system.

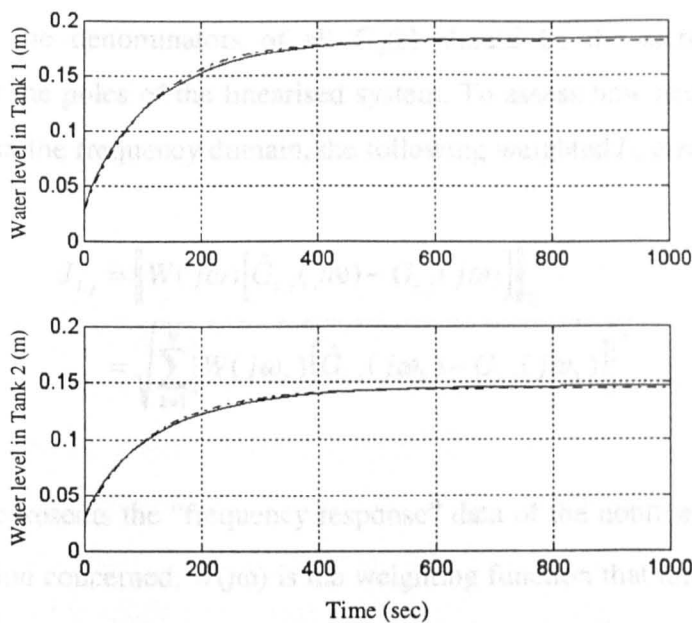


Figure 3.16 Output responses of the linearised model {—} and the nonlinear system { - - - }

### 3.3.2 Frequency Domain $L_2$ Linearisation

Although the time domain evolutionary  $L_2$  system linearisation techniques developed in Section 3.3.1 has offered good performances with accurate linearised model, the linearisation frame work requires a large number of samples (Pintelon *et al.*, 1994). To overcome the limitation, the multivariable frequency domain evolutionary  $L_2$  linearisation technique based on the system input-output behaviour is developed in this section. The laboratory coupled liquid tank system shown in Figure 3.6 is to be linearised using the hybrid EA developed in Chapter 2. If this two-input and two-output nonlinear system can be linearised, a transfer function matrix model given by:

$$\hat{G}(s) = \begin{bmatrix} \hat{G}_{11}(s) & \hat{G}_{12}(s) \\ \hat{G}_{21}(s) & \hat{G}_{22}(s) \end{bmatrix} \quad (3.22)$$

may be used to describe the linearised system, where  $i, j \in \{1, 2\}$  and every transfer function element is of the form:

$$\hat{G}_{ij}(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_0} \quad (3.23)$$

It is noted that the denominators of all  $G_{ij}(s)$  should be the same, determining the characteristics or the poles of the linearised system. To assess how accurate the linearised model performs in the frequency domain, the following weighted  $L_2$  error norm is used,

$$\begin{aligned} J_{i,j} &= \left\| W(j\omega) [\hat{G}_{i,j}(j\omega) - G_{i,j}(j\omega)] \right\|_2 \\ &= \sqrt{\sum_{k=1}^N \left| W(j\omega_k) [\hat{G}_{i,j}(j\omega_k) - G_{i,j}(j\omega_k)] \right|^2} \end{aligned} \quad (3.24)$$

where  $G_{i,j}(j\omega_k)$  represents the “frequency response” data of the nonlinear system under the operating condition concerned;  $W(j\omega)$  is the weighting function that allows fitting errors in some chosen parts of the frequency range to have particular emphasis; and  $N$  is the number

of data points used. The objective here is to obtain an optimal  $\hat{G}(s)$  of multiple parameters that results in the minimum linearisation error as defined by:

$$J = \sum_{j=1}^2 \sum_{i=1}^2 J_{i,j} \quad (3.25)$$

If it is required, however, the diagonal and non-diagonal elements may be weighted separately. The complexity in obtaining solutions to multimodal problems like this has prevented the use of analytical and conventional numerical means (Tan *et al.*, 1996). One possible way to obtain numerical solutions to such MIMO system linearisation problems would be to search exhaustively for an optimal answer. This is, however, practically impossible, as an enumerative algorithm requires exponential, as opposed to polynomial, search time and will thus easily break down due to the high parametric dimensionality of this type of problems.

### Linearisation from a Nonlinear Model

Without loss of generality, consider the nonlinear system in Equation (3.17) to be linearised to a causal first and second-order model given by Equations (3.22) and (3.23) with  $n = 1, 2$  and  $m \leq 1$ . To reflect practical applications as discussed above, the linearised model is to provide the best fit for a pre-specified operating region around the operating point. This means two DC voltages are first applied to set the steady-state operating (and equilibrium) points and then two pseudo random binary sequences (PRBS) voltages with, for example,  $\pm 5\%$  amplitude relative to the set voltages are applied.

In Equation (3.24) and Equation (3.25), the task of minimising the errors of transfer is reduced to the task of minimising the output errors. In this study, the interested frequency ranges from  $4 \times 10^{-3}$  rad/s to  $4 \times 10^{-2}$  rad/s, as the plant is relatively sluggish. In the simulation, the operating point of Tank 1 is set at 18 cm and that of Tank 2 at 15 cm using step input voltages. Time domain simulation on the differential equations given by Equation (3.17) has been performed, using the fourth-order Runge-Kutta numerical integration method. By holding the input to tank 2 constant, the PRBS signal is added to Input 1 and both outputs are collected for evaluating  $W_1(j\omega)G_{11}(j\omega)$  and  $W_1(j\omega)G_{21}(j\omega)$ .

Then this procedure is repeated for Input 2. In order to acquire the frequency response data of the cross transfers in the nonlinear MIMO system, the collected time domain output and input data are converted to spectra using Matlab (MathWorks, 1992) function Fast Fourier Transforms (FFT) for the frequency range between  $3 \times 10^{-3}$  rad/s and  $5.5 \times 10^{-2}$  rad/s, over-covering the interested points. By dividing the magnitudes of the output spectra with those of the corresponding input, the gains of the point-linearised system are obtained. The phases of the system can also be obtained by subtracting the phases of the output from those of the inputs.

In the evolutionary linearisation program, different candidate parameter sets of the linear models in the population results in different  $G_{i,j}(j\omega_k)$  with different linearisation quality. The linearisation program was executed to evolve for 150 generations of 100 candidate solutions to Equation (3.25). The best linearised first and second-order models and its RMS errors obtained at the end of evolution are given in Table 3.3.

Table 3.3 Linearised models of the coupled liquid-level system

Orders	Transfer function matrices	$[J_{\text{RMS}}(\theta)]$
1 (variable)	$\frac{1}{56.5s + 0.743} \begin{bmatrix} 0.955 & 0.612 \\ 0.633 & 0.668 \end{bmatrix}$	$\begin{bmatrix} 0.0833 & 0.0649 \\ 0.0638 & 0.0562 \end{bmatrix}$
2	$\Delta^{-1}(s) \begin{bmatrix} 4.1041s + 0.7688 & 0.4805 \\ 0.4945 & 5.3153s + 0.5305 \end{bmatrix}$ $\Delta(s) = s^2 + 49.05s + 0.5385$	$\begin{bmatrix} 0.1126 & 0.0641 \\ 0.0643 & 0.0475 \end{bmatrix}$

The frequency response of the second order linear model is depicted by the dotted lines in Figure 3.17. For comparison, frequency response of the FFT transformed data is also plotted in Figure 3.17. It can be seen that both magnitude and phase are fitted accurately, although the minimisation objectives given by Equation (3.24) and Equation (3.25) are mainly focused on magnitude.

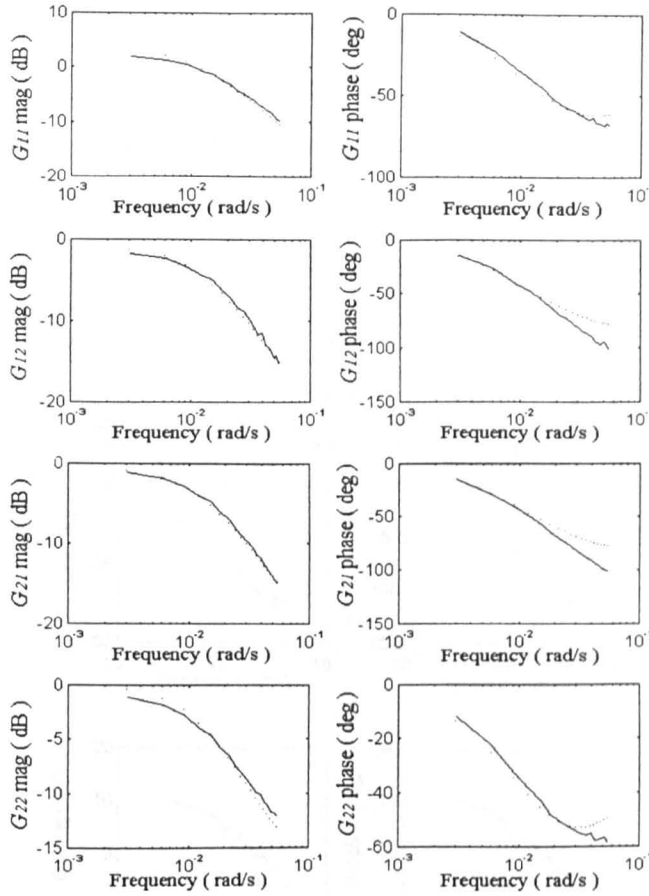


Figure 3.17 Frequency responses of the second order linear model (dotted) and the FFT transformed data (solid)

### Linearisation Directly from Plant Response Data

To further validate the hybrid EA linearisation technique, similar procedures that used in the previous section have been applied to obtain the frequency response of the physical coupled liquid system shown in Figure 3.6. The evolutionary linearisation program was run for 150 generations with a population of 100. The best linearised second-order model obtained at the end of evolution is given by:

$$G(s) = \frac{1}{\Delta} \begin{bmatrix} 13.61s + 2.88 & 1.18018 \\ 1.8338 & 6.006s + 1.9399 \end{bmatrix} \quad (3.26)$$

where  $\Delta = s^2 + 80.08s + 0.4965$ . For comparison, frequency response of the real physical coupled liquid-level system and the linearised model are plotted in Figure 3.18, which indicates a satisfactory fitting over the whole relevant frequency range.

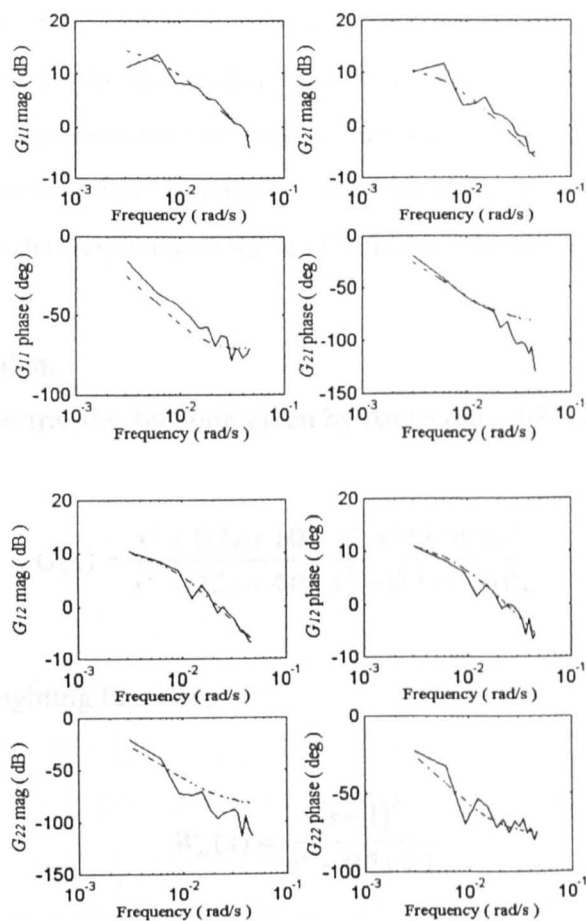


Figure 3.18 Frequency responses of the linearised (dotted) and the real physical (solid) system

### 3.4 $L_\infty$ Model Reduction by Evolution

Given a transfer function  $G(s)$ , the objective of frequency-weighted  $L_\infty$  norm model reduction is to find a transfer function  $G_r(s)$  such that the cost  $\left\| W_\alpha(s)[G(s) - G_r(s)] \right\|_\infty$  is minimised, given a frequency weighting function  $W_\alpha(s)$ . The following lemma provides a lower bound for an  $L_\infty$  norm approximation (Glover, 1984).

**Lemma** Given an  $m^{\text{th}}$  order transfer function  $G(s)$ , there is an  $r^{\text{th}}$  order  $G_r(s)$  such that  $\sigma_{r+1} \leq \|G - G_r\|_{\infty}$ , where  $\sigma_{r+1}$  is the  $(r+1)^{\text{th}}$  Hankel Singular Values of  $G$ .

Unlike Hankel norm model reduction, the lower bound is not necessarily achievable in the  $L_{\infty}$  norm-based model reduction. In general, the computation for an optimal solution to the above model reduction problem is still an open issue. These problems are difficult to solve using conventional optimisation techniques that require a differentiable error energy function and appropriate initial conditions. The hybrid EA detailed in Chapter 2 is thus applied here to solve the frequency-weighted  $L_{\infty}$  model reduction problems.

### A Benchmark Problem

Consider the 4<sup>th</sup> order transfer function given by Anderson (1986) and Zhou (1995):

$$G(s) = \frac{s^2 + 0.2s + 1.01}{s^2 + 0.2s + 4.04} \frac{s^2 + 0.2s + 9.01}{s^2 + 0.2s + 16.02} \quad (3.27)$$

with a frequency weighting function,

$$W_{\alpha}(s) = \frac{(s-1)^2}{s^2 - 0.2s + 1} \quad (3.28)$$

The hybrid EA has been run for 150 generations with a population size of 100. The optimally identified second and third order reduced models at the end of evolution are

$$G_{r,2nd}(s) = \frac{0.96s^2 + 2.7448s + 2.31}{s^2 + 0.459s + 17.146} \quad (3.29)$$

$$G_{r,3rd}(s) = \frac{4.2526s^3 + 8.033s^2 + 81.802s + 1.216}{s^3 + 12.2327s^2 + 19.4596s + 200.3} \quad (3.30)$$

Table 3.4 gives the model reduction errors of the hybrid EA based method and the various well known methods in the literature. It can be seen that the evolutionary and learning

technique outperforms others, yielding the smallest  $L_\infty$  norm errors for both the 2<sup>nd</sup> and 3<sup>rd</sup> order reductions.

Table 3.4 The model reduction  $L_\infty$  norm errors in  $\| W_\alpha(s)[G(s) - G_r(s)] \|_\infty$

Identified Model Order	2 <sup>nd</sup>	3 <sup>rd</sup>
<i>Lower Bounds (HSV)</i>	2.704	2.527
<i>Latham &amp; Anderson (1986)</i>	20.08	11.94
<i>Chiang &amp; Safonov (1992)</i>	11.71	6.303
<i>Zhou (1995) (Algorithm I)</i>	4.827	8.20
<i>Zhou (1995) (Algorithm II)</i>	4.822	3.946
<i>Hybrid EA</i>	4.517	3.789

In Table 3.4, HSV stands for the  $(r+1)^{th}$  Hankel singular value of  $W_\alpha(s)G(s)$ , which is the theoretical lower bound of the error given by the Lemma. The frequency-weighted errors for both the second and the third order reduced models are shown in Figure 3.19. Responses shown that the hybrid EA gave a small and tight  $L_\infty$  error bounds for both the 2<sup>nd</sup> and 3<sup>rd</sup> order reductions. Figure 3.19 also indicates that the choice of frequency weighting function of Equation (3.28) is inappropriate to this model reduction problem, as it produces a few undesirable error peaks between the frequency range of 1 rad/s and 10 rad/s. Due to the purpose of comparison, the weighting function was not altered in this ‘benchmark’ example.



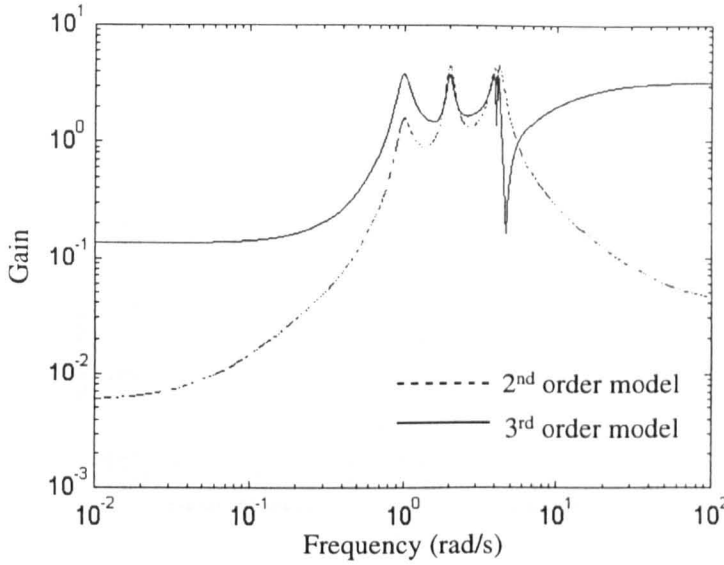


Figure 3.19 Gain plots of the weighted model reduction errors

### 3.5 $L_\infty$ Linearisation: Model Reduction from an Infinite-Order

In this Section, the equivalent “infinite order” of the nonlinear coupled liquid-level system shown in Figure 3.6 is to be reduced using the evolution based  $L_\infty$  model reduction technique. Here, steps with additive PRBS inputs were used to excite the nonlinear system around an operation point at 18 cm of Tank 1 and 15 cm of Tank 2. Frequency domain data of  $G_{ij}(s)$  of Equation (3.22) was obtained by converting the collected time domain output and input data to spectrum using FFT as described in Section 3.3.2. In order to obtain a reduced linearised model that offers smallest  $L_\infty$  norm errors, the hybrid EA program was run for 150 generations with a population of 100. The linearised second-order model obtained at the end of evolution is given by

$$G_r(s) = \frac{1}{\Delta} \begin{bmatrix} 20s + 2.9 & 1.95 \\ 2.53 & 1.07s + 2.15 \end{bmatrix} \quad (3.31)$$

where  $\Delta = s^2 + 87.19s + 0.58$ . The frequency response of the real physical nonlinear system and the linearised model are plotted in Figure 3.20. It can be seen that the linearised model provides a satisfactory fitting over the relevant frequency range.

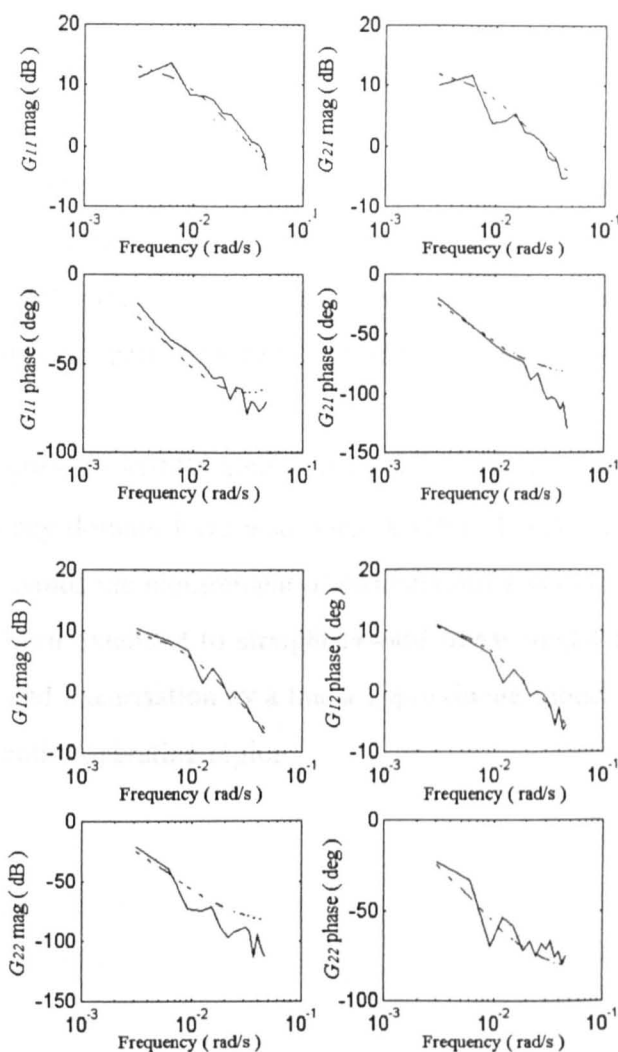


Figure 3.20 Frequency responses of the linearised (dotted) and the real physical (solid) system

3.6 Summary

This Chapter has developed a Boltzmann learning enhanced evolutionary algorithm technique for  $L_2$  norm and  $L_\infty$  norm based model reduction and linearisation. The EA based reduction method is generic and applicable to both discrete and continuous-time systems by minimising the  $L_2$  or  $L_\infty$  norms between the original and reduced models. It is applicable to both SISO and MIMO systems in both time and frequency domains and can be easily

extended to  $L_1$  norm based reductions. The technique also provides a tighter  $L_\infty$  error bound than existing methods for the  $L_\infty$  model reduction problem.

Enabled by a control gene for structural switch, this hybrid EA based technique is capable of simultaneously recommending both an optimal order number and corresponding parameters. The model reduction examples show that this indirectly guided optimising method provides a superior performance to that of existing methods.

Evolutionary techniques for MIMO nonlinear  $L_2$  and  $L_\infty$  norm system linearisation in both the time and frequency domain have also been developed. Plant I/O data is used for the linearisation which avoids the requirement of differentiating system nonlinearities. Further, the approach have been extended to straightforward linear model approximation, inverse model linearisation and linearisation by a linear approximate-model network, which allows linearisation for an entire operating region.

## Chapter 4

# EVOLUTIONARY SYSTEM IDENTIFICATION AND MODELLING

### 4.1 The System Identification and Modelling Problem

System identification and modelling techniques are used in many fields in order to obtain an accurate mathematical model of a dynamic system, by fitting from the observed input-output data of the physical system (Ljung, 1987; Söderström and Stoica, 1989). The fitting accuracy is usually measured by a performance index  $f(m_i)$ . Here  $m_i \in M$  represents the structure and parameters of a candidate model within the space  $M$  of all possible models. The task of system identification and modelling is to find a  $m_0$  such that  $f(m_0) = \sup f(M)$ . The choice of the performance measure usually depends on the purpose that the identified model is to be served and of how easily such a performance index can be optimised over the space  $M$ . For example, the traditional preference of  $L_2$  norm for maximum likelihood estimation method is certainly due in part to its computational efficiency and problem formulation. Similarly, the  $L_\infty$  norm for worst-case system identification is used for its compatibility to the framework of robust control applications. Other norms such as  $L_1$  norm may also be employed if minimum absolute error measure is desired.

As addressed in Chapter 1, conventional optimisation techniques such as least mean-squares (LMS) or maximum likelihood estimates require a differentiable performance index  $f$  or a smooth search space. This condition cannot always be satisfied in practical applications, where the index may not be ‘well-behaved’ because of noisy data. Even when the condition is satisfied, the conventional techniques may only lead to a local optimum if the search space is multidimensional. The problem will become more complicated if the plant is a multiple-input and multiple-output (MIMO) system.

In this Chapter, the hybrid EA detailed in Chapter 2 is applied to solve the system identification and modelling problems. This hybrid EA has shown better performance and accuracy than other search methods such as downhill simplex method and simulated

annealing (SA), upon a comparison in the benchmark problem studied in Chapter 2. Although search methods and non-search methods are not under the same category of optimisation techniques, many results in this thesis obtained from the hybrid EA search method are compared with that of non-search gradient-guided LMS method, since LMS methods are well established for the problems of system identification.

Since EAs only rely on performance evaluations and require no derivative information, different cost function with  $L_1$ ,  $L_2$  and  $L_\infty$  norm can be easily adopted as a performance measure without major modification to its problem formulation. This has made the choice of performance measure in an EA more flexible than conventional LMS approaches. By adopting an appropriate performance measure, identification bias due to process or measurement noise could be avoided in an EA based system identification (Sharman, *et al.*, 1995). Problems such as flat landscape or multiple solutions of the cost function could also be overcome if niche induction technique as studied in Section 2.5.1 is accommodated in the EA.

Here, the evolutionary system identification and modelling techniques to be developed are aimed at overcoming tractability difficulties encountered in conventional methods in two ways. One is that an EA is an indirect search method and requires no *a-priori* directional guidance. The other is that an EA conducts multiple searches in parallel by a population of individuals with effective exchange of co-ordinate information (parameters) in the search.

Evolutionary  $L_2$  norm based system identification of a benchmark discrete-time dynamic system and a DC servo-system are to be studied in Section 4.2. The benchmark system is represented by an ARMAX (Auto-Regressive Moving Average model with eXogenous inputs) black-box model in the presence of white noise, and the DC servo-system is represented by an ODE based clear-box model. Transportation delay is also accommodated in the EA based identification for an more accurate representation of the system. Results obtained from the EA and from conventional least mean-squares (LMS) based techniques will be compared. Evolutionary  $L_\infty$  norm system identification for robust control applications is studied in Section 4.2. The method is to be uniformly applicable to both continuous and discrete time systems. Using this technique, both a globally optimised nominal model and an error boundary function for additive and multiplicative uncertainties

are to be evolved. Evolutionary nonlinear system identification of physical law based clear-box models is studied in Section 4.3. Section 4.4 develops a novel grey-box modelling methodology for nonlinear systems. The technique is to utilise the clear-box dominated global structures, with local black-boxes to accommodate unmeasurable nonlinearities or 'coefficients'. Evolution based closed-loop system identification for both linear and nonlinear systems will be studied in Section 4.5. A summary is given in Section 4.6.

## **4.2 Efficient and Accurate Evolution Based Linear System Identification**

In control engineering, a practical system is usually described by a set of ordinary differential equations (ODEs), in a clear-box model structure derived from the physical mechanism of the system. The parameters of these ODEs often have a clear physical representation of the system. Although parameters of linear black-box models have no physical meanings of a system, the mappings from linear black-box models to clear-box ODEs are bijective. This implies evolutionary system identification techniques for both clear-box and black-box models are the same for a linear system. In general, clear-box models are used for time-domain ODEs while black-box models for frequency domain transfer functions. In either case, only independent parameters of the models can be identified.

### **4.2.1 Evolutionary $L_2$ System Identification**

#### **4.2.1.1 ARMAX Based Black-Box Model**

In general,  $L_2$  norm or  $\sqrt{N}$  times the root-mean square (RMS) error, is employed as a performance measure for identification accuracy. Using this criterion, a candidate model  $m_i$  that provides a quadratic errors over the space of  $M$  is to be obtained. When nothing is known regarding the measurement errors,  $L_2$  norm is preferred as the measure of identification accuracy since it does not impose any statistical assumptions about the measured errors (Beck and Arnold, 1977; Ljung, 1987). In this Section, the hybrid EA detailed in Chapter 2 is applied to identify a linear discrete-time system. The system can be

described by an auto-regressive moving average exogenous (ARMAX) black-box model as given by:

$$A(q^{-1})y(t) = B(q^{-1})u(t - T) + C(q^{-1})d(t) \quad (4.1)$$

where  $d(t)$  is a white noise sequence with zero mean and unit variance;  $u(t)$  and  $y(t)$  are the system input and output;  $q$  is the forward shift operator and  $T$  is the system transport delay.  $A(q^{-1})$ ,  $B(q^{-1})$  and  $C(q^{-1})$  are polynomials of the delay operator  $q^{-1}$ . A benchmark system that is often used to test various identification methods is given by (Kristinsson and Dumont, 1992; Ljung, 1987; Söderström and Stoica, 1989):

$$y(t) = b_0u(t-5)+b_1u(t-6)-a_0y(t-1)-a_1y(t-2)+c_0d(t)+ c_1d(t-1)+ c_2d(t-2) \quad (4.2)$$

where  $T = 5$ ;  $a_0 = -1.5$ ;  $a_1 = 0.7$ ;  $b_0 = 1.0$ ;  $b_1 = 0.5$ ;  $c_0 = 1.0$ ;  $c_1 = -1.0$  and  $c_2 = 0.2$ . The objective here is to use the hybrid EA to identify  $T$  and the parameters  $a_0$ ,  $a_1$ ,  $b_0$  and  $b_1$  of Equation (4.2) in the presence of noise. The excitation input  $u(t)$  to be used here is a step added by a pseudo random binary sequence (PRBS) with 10% of the step amplitude (Tan *et al.*, 1995). Figure 4.1 shows the input and output of the system.

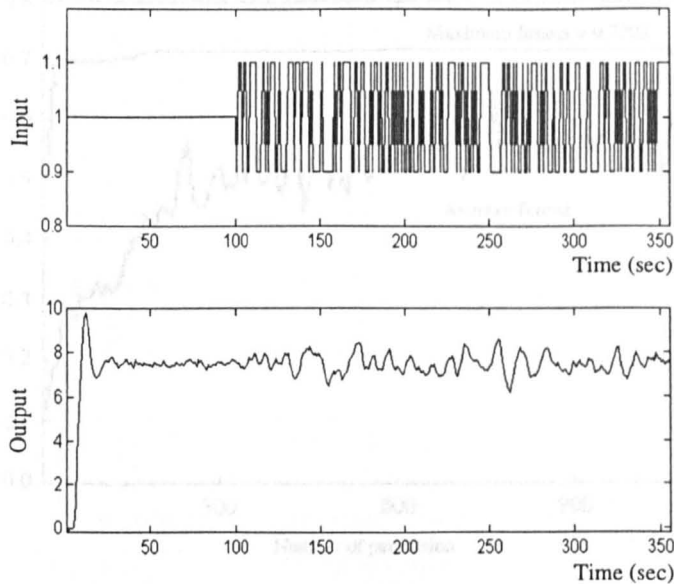


Figure 4.1 Input with PRBS and output with noise of a system

A generalised inverse performance index is given by

$$f^{-1} = \|w(t) \cdot e(t)\| \tag{4.3}$$

where  $e(t)$  is the error between the actual and modelled outputs and  $w(t)$  is a weighting function to be used when some time history needs to be emphasised. Here, no weighting is used and the  $L_2$  norm is employed as in the general cases. The searching range for the parameters in  $A(q^{-1})$  and  $B(q^{-1})$  in this example is targeted for  $[-5, 5]$  but may be varied depending on *a-priori* knowledge of the system parameters. Here, each parameter is encoded by 4 digits, 3 of which represent the relative value of the candidate parameter for the range selected by the first digit. The transport delay term  $T$  is coded as a single digit in the multiples of the sampling period.

A pure and the hybrid evolutionary algorithm programmed in Turbo Pascal are applied to this problem, with a population size of 300 and 150 respectively. The initial populations of the parameter sets are generated randomly. Figure 4.2 shows both the average and the highest fitness in every generation of the two algorithms. It can be seen that the hybrid algorithm has yielded faster convergence and better accuracy.

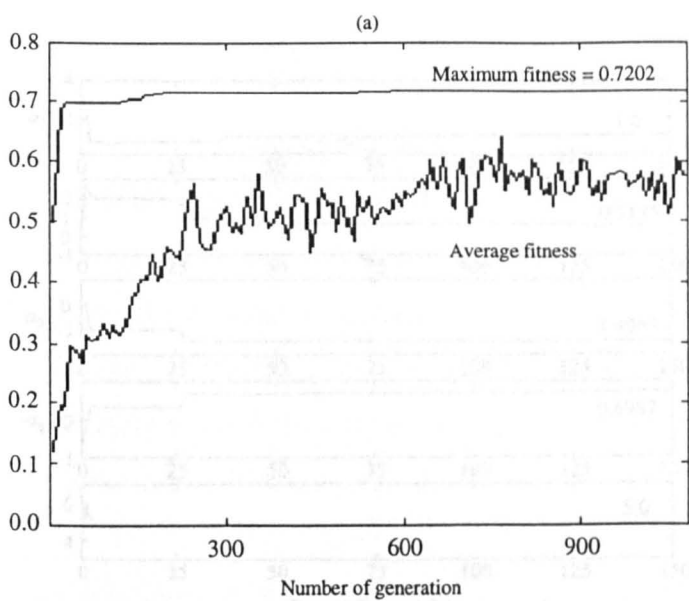


Figure 4.3 Convergence trace of the best parameter set in the hybrid algorithm



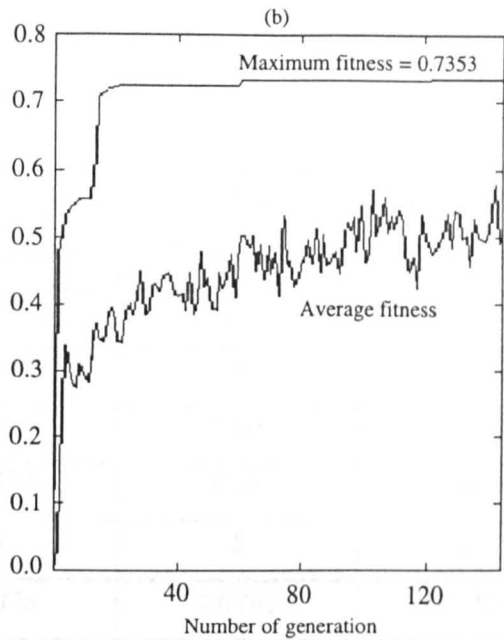


Figure 4.2 Convergence trace of normalised fitness: (a) Pure EA; (b) Hybrid EA

A convergence plot of the five parameters evolved by the hybrid algorithm is shown in Figure 4.3. It can be seen that, even with the presence of noise in the output data set, the identification converges rapidly from a random starting point. The parameters and RMS error resulting from the pure EA and the hybrid EA at the end of 1100<sup>th</sup> and 150<sup>th</sup> generation, respectively, are given in Table 4.1.

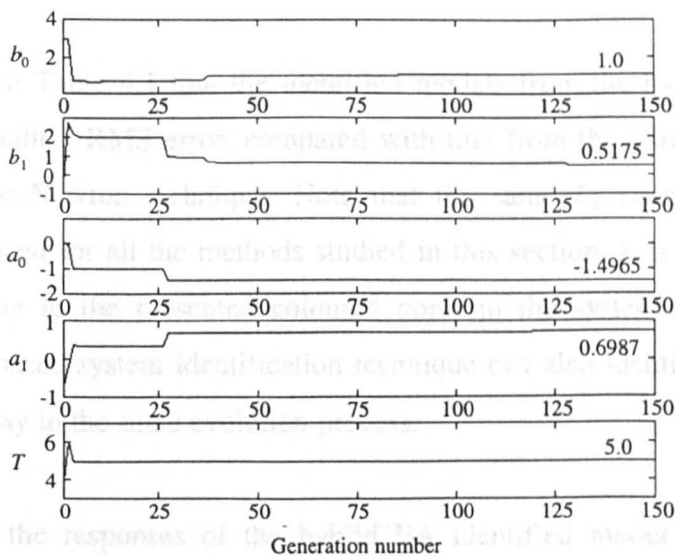


Figure 4.3 Convergence trace of the best parameter set in the hybrid EA

Table 4.1 Set of parameters obtained from the hybrid EA, pure EA and Gauss-Newton methods

Parameters in model	Hybrid EA (150 generations)	Pure EA (1100 generations)	Kristinsson & Dumont	MATLAB (Gauss-Newton)
$a_0 = -1.5$	-1.4965	-1.47	-1.496	-1.4894
$a_1 = 0.7$	0.6987	0.68	0.696	0.6888
$b_0 = 1.0$	1.0	1.03	1.039	0.921
$b_1 = 0.5$	0.5175	0.56	0.4426	0.5733
$T = 5$	5	5	5	<i>A-priori</i>
<i>RMS error</i>	0.0078	0.0762	0.0921	0.0282

For comparison, the iterative Gauss-Newton technique is also applied here, using the Matlab (MathWorks, 1992) System Identification Toolbox (Ljung, 1992). The Gauss-Newton vector is bisected up to ten times until a lower value of the criterion is found. The initial conditions for the iterative search are constructed in a four stage LS-IV algorithm accommodated with the toolbox. The transport delay  $T$  is assumed known *a-priori* and is incorporated in the model, because delay terms cannot be identified using the System Identification Toolbox. The parameters and the RMS error obtained are also shown in Table 4.1.

It can be seen from Table 4.1 that the identified models from the hybrid EAs are most accurate with a smallest RMS error, compared with that from the pure EA and from the conventional Gauss-Newton technique. Note that the same  $L_2$  norm cost function of Equation (4.3) is used for all the methods studied in this section. It is thus all the results here are biased due to the presented coloured noise in the system. Apart from better accuracy, the EA based system identification technique can also identify both the system parameters and delay in the same evolution process.

Figure 4.4 shows the responses of the hybrid EA identified model against the actual ARMAX model without the presence of the noise, using the same input data set, confirming the superior performance of the EA. It should be noted that, however, pole-zero

identifications by an EA will usually yield a biased estimate for zeros (Kristinsson and Dumont, 1992). Further, it is difficult to determine the searching ranges for the poles and zeros if the system to be identified is unstable or is of non-minimal phase.

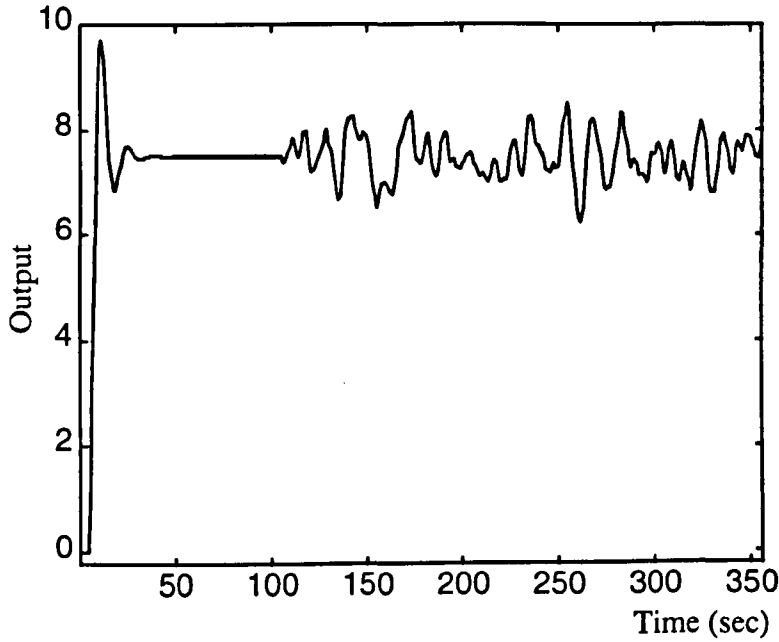


Figure 4.4 Responses of the system { — } and the hybrid EA identified model { - - - } without the presence of colour noise

#### 4.2.1.2 ODE Based Clear-Box Model

This section applies the hybrid EA to the identification of an LTI DC servo-system. A typical ODE defining the open-loop servomotor system with field control is given by:

$$\frac{d^2\omega}{dt^2}(t-T) + \left(\frac{JR+LB}{LJ}\right)\frac{d\omega}{dt}(t-T) + \left(\frac{RB}{LJ}\right)\omega(t-T) = \left(\frac{K_T}{LJ}\right)v_{in} \quad (4.4)$$

where  $v_{in} \in [-5V, 5V]$  is the input field control voltage;  $T$  is the transportation delay of the system;  $K_T$  in Nm/A the torque constant,  $R$  in  $\Omega$  the resistance of the motor winding.  $L$  in H the inductance,  $B$  in Nms the friction coefficient of the shaft, and  $J$  in  $Kgm^2$  the moment of inertia of the load and the machine.

As addressed easier, only independent parameters in a clear-box model can be identified. The identification task here is thus to use the hybrid EA to identify  $T$  and the combined ODE coefficients in Equation (4.4). Step response data used for the identification is shown by the solid line in Figure 4.5. The inverse performance index given by Equation (4.3), and  $L_2$  norm are used here. After running the hybrid EA for 50 generations with a population size of 50, the evolved combined coefficients in Equation (4.4) and RMS error are shown in Table 4.2. To compare, step response of the EA identified model is also shown by the dotted line in Figure 4.5. Clearly, the model has performed a good fitting to the DC servo-system response.

Table 4.2 The hybrid EA identified independent parameters

Independent Parameters in Equation (4.4)	Hybrid EA Identified Parameters
$\frac{JR + LB}{LJ}$	7.13
$\frac{RB}{LJ}$	6.33
$\frac{K_T}{LJ}$	4.32
$T$	0.04
RMS error	0.00664

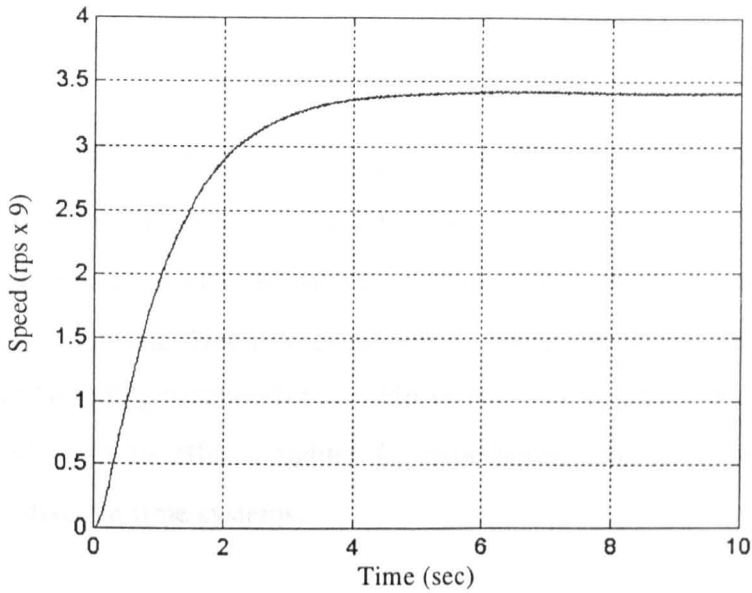


Figure 4.5 Responses of the DC servo-system { — } and the EA identified model { - - - }

the system  $G(s)$  is to be identified using a nominal model  $\hat{G}(s)$  and an uncertainty bound  $\Delta G(s)$  such that

### 4.2.2 Evolutionary $L_\infty$ System Identification

With rapid development in robust control theory and algorithms, system identification techniques which are compatible with the frequency domain mini-max design framework have received increasing attention. Such techniques should identify a nominal model that best matches the given experimental data in terms of the  $L_\infty$  norm and provides an uncertainty bound, or an upper limit, of the identification error.

Figure 4.6

Recently, many useful methods for robust control oriented system identification have been developed in the frequency domain (Gu and Khargonekar, 1992; Helmicki *et al.*, 1991, 1993). However, these methods have failed to provide information on the uncertainty bounds and the resulting nominal models tend to be of high order in general. A few time domain set-membership (Kosut *et al.*, 1992) and interpolation (Zhou and Kimura, 1994) approaches manage to provide relatively low orders, but an accurate uncertainty bounding function must be given *a-priori*, which is hardly feasible in practice. The set-membership based  $L_\infty$  system identification approach proposed by Sugie and Tanai (1995) succeeds in identifying both a nominal model and a uncertainty bounding function. However, it

requires the nominal model and the bounding function to be linear in parameterisation and good initial condition is required.

This section highlights problems in  $L_\infty$  system identification for robust control in the frequency domain and develops a hybrid EA based technique to solve  $L_\infty$  identification problems. In the context of identification for robust control, the  $L_\infty$  norm is adopted as the cost function for modelling (Helmicki *et al.*, 1991). Using this technique, both the nominal model and error bounding function for an additive and multiplicative uncertainty model are to be optimised. It is to offer a tighter  $L_\infty$  error bound and to be applicable to both continuous and discrete-time systems.

#### A. Additive Uncertainty Model for Robust Control Application

Suppose the plant  $G(s)$  is to be identified using a nominal model  $G_r(s)$  with an additive uncertainty as described by (Doyle and Stein, 1981; Sugie and Tanai, 1995):

$$G(s) = G_r(s) + \Delta(s)W(s) \quad (4.5)$$

where  $\Delta(s)$  is unknown but is bounded as given by

$$\|\Delta(s)\|_\infty \leq 1 \quad (4.6)$$

Define

$$H(s) = G(s) - G_r(s) \quad (4.7)$$

to represent the error transfer function resulting from the approximate model.

The  $L_\infty$  identification objective is to find an optimal  $G_r(s)$  such that the cost function

$$\begin{aligned} J_G &= \|W_\alpha(s)H(s)\|_\infty = \|W_\alpha(s)\Delta(s)W(s)\|_\infty \\ &\leq \|W_\alpha(s)W(s)\|_\infty \end{aligned} \quad (4.8)$$

is minimised given a frequency weighting function  $W_\alpha(s)$ . Although  $J_G$  is always bounded by  $\|W_\alpha(s)W(s)\|_\infty$ , the tighter the bound, the more accurate the nominal model can reflect the actual plant dynamics and, thus, the higher the potential performance of the robust controller may offer. Therefore, in the identification, another objective is to find an optimal  $W(s)$  such that the cost (Sugie and Tanai, 1995):

$$J_w = \left\| W_\beta(s)[W(s) - H(s)] \right\|_\infty \quad (4.9)$$

is minimised under the constraint:

$$\left\| H(s)W^{-1}(s) \right\|_\infty \leq 1 \quad (4.10)$$

derived from Equations (4.5), (4.6) and (4.7). Here,  $W_\beta(s)$  is also a frequency weighting function. The introduction of such a scaling function allows fitting errors in chosen parts of the frequency range to have a particular emphasis if needed.

Note that, it is impossible to minimise both  $J_G$  and  $J_w$  for an infinite number of frequency points, as the true plant is unknown and the identification is carried out from the plant frequency response data. However, minimising the approximate cost

$$J_G = \max_{k \in \{1, \dots, n\}} |W_\alpha(j\omega_k)H(j\omega_k)| \quad (4.11)$$

over the interested frequency range is possible and is adopted in practice (Zhou and Kimura, 1993). Similarly, based on the obtained nominal model, the uncertainty bounding function is to be determined by minimising

$$J_w = \max_{k \in \{1, \dots, n\}} \left| W_\beta(j\omega_k)[W(j\omega_k) - H(j\omega_k)] \right| \quad (4.12)$$

subject to the constraint:

$$\max_{k \in \{1, \dots, n\}} |H(j\omega_k)W^{-1}(j\omega_k)| \leq 1 \quad (4.13)$$

where  $n$  is a finite number of points covering the frequency range concerned.

### B. Multiplicative Uncertainty Model for Robust Control Application

A multiplicative uncertainty model of a continuous-time system is represented by (Doyle *et al.*, 1992):

$$G(s) = G_r(s)[1 + \Delta(s)W(s)] \quad (4.14)$$

where  $W(s)$  is now the multiplicative uncertainty bounding function. Again, the same  $L_\infty$  cost of  $J_G$  of Equation (4.11) is used to identify an optimal nominal model  $G_r(s)$ . In this case, however, the corresponding multiplicative uncertainty bounding function  $W(s)$  can be obtained by minimising the cost (Doyle *et al.*, 1992):

$$J_W = \max_{k \in \{1, \dots, n\}} \left| W_\beta(j\omega_k) \left\{ W(j\omega_k) - \left[ \frac{G(j\omega_k)}{G_r(j\omega_k)} - 1 \right] \right\} \right| \quad (4.15)$$

subject to the constraint:

$$\max_{k \in \{1, \dots, n\}} \left| \left[ \frac{G(j\omega_k)}{G_r(j\omega_k)} - 1 \right] W^{-1}(j\omega_k) \right| \leq 1 \quad (4.16)$$

### C. $L_\infty$ System Identification for Discrete-Time Systems

In the discrete-time, the additive and multiplicative uncertainty model descriptions are:

$$G(z) = G_r(z) + \Delta(z)W(z) \quad (4.17)$$

and

$$G(z) = G_r(z)[1 + \Delta(z)W(z)] \quad (4.18)$$



respectively. Again,

$$\|\Delta(z)\|_{\infty} \leq 1 \quad (4.19)$$

All derivations in (A) and (B) hold by replacing  $j\omega$  with  $e^{j\omega T}$  and by changing the argument  $s$  to  $z$ . Note that, however, the  $L_{\infty}$  norm for a discrete-time system is given by:

$$\|G(z)\|_{\infty} = \sup_{|\omega T| \leq \pi} |G(e^{j\omega T})| \quad (4.20)$$

where  $T$  is the sampling period.

### Identification for a Discrete-Time Heat Exchanger

Study a 4<sup>th</sup> order discrete plant of an industrial heat exchanger (Golten and Verwer, 1991) given by:

$$G(z) = \frac{0.049(z + 0.72)}{z^2(z - 0.607)^2} \quad (4.21)$$

with a sampling period of  $T = 7.5$  s. It is to be modelled by a discrete-time additive uncertainty model with a first order uncertainty bounding function. The discrete-time frequency weighting functions in this case are chosen as  $W_a(z) = 1$  and

$$W_{\beta}(z) = \frac{7.5z - 4.3}{z - 0.34} \quad (4.22)$$

The EA detailed in Chapter 2 is applied here with a population of 100. After running the EA for 100 generations, the evolved second and third order discrete nominal models are given by:

$$G_{r,2nd}(z) = \frac{-0.0225z + 0.0628}{z^2 - 1.5856z + 0.668} \quad (4.23)$$

$$G_{r,3rd}(z) = \frac{-0.00027z^2 - 0.004z + 0.067}{z^3 - 1.3954z^2 + 0.5155z - 0.00335} \quad (4.24)$$

and their uncertainty bounding functions are:

$$W_{2nd}(z) = \frac{0.0635}{z} \quad (4.25)$$

$$W_{3rd}(z) = \frac{0.01923}{z} \quad (4.26)$$

respectively. Frequency responses of the true discrete plant and the discrete nominal models are plotted in Figure 4.6. It can be seen that the proposed hybrid evolutionary technique gave excellent identification results, with a good fitting over the frequency range concerned. This also means that equivalent or reduced-order models can be obtained with a known error bound, which is applicable to the framework of robust control.

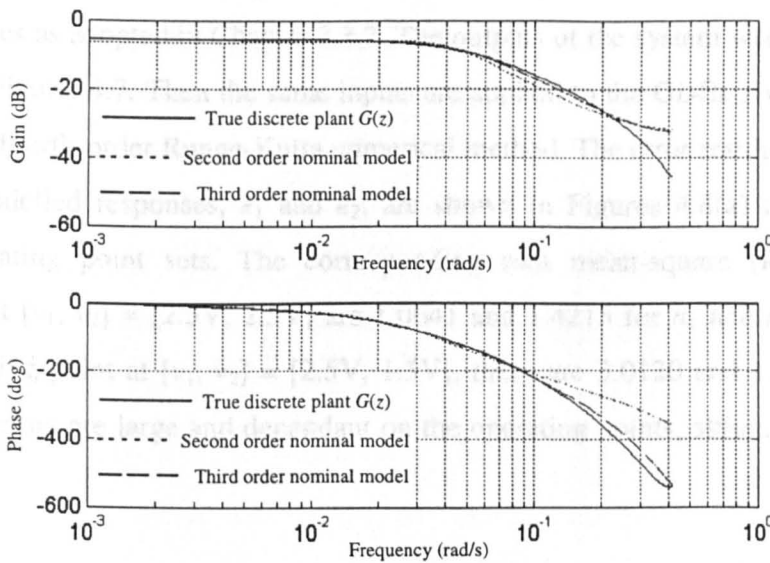


Figure 4.6 Frequency responses of the true discrete plant and the discrete nominal models

### 4.3 Efficient and Accurate Evolution Based Nonlinear Clear-Box Identification

#### 4.3.1 A Coupled Nonlinear System and Measured Clear-Box Model

The nonlinear coupled twin-tank hydraulic system shown in Figure 3.6 will be experimented in this Section. A clear-box model of the system is given in Equation (3.17). Based on the manufacturer's specification and further physical measurements at steady-state with both inputs at 2.5 V, the system parameters are obtained as

$$\frac{C_1 a_1}{A} = 0.0021 \quad \frac{Q_1}{A} = \frac{Q_2}{A} = 0.07 \text{ cm s}^{-1} \text{V}^{-1} \quad \frac{C_2 a_2}{A} = 0.0024 \quad H_3 = 3 \text{ cm}$$

To study this clear-box model obtained by the physical law and actual measurements, steps plus delayed small pseudo random binary sequences (PRBS) were first used to excite the physical system at the operating levels of  $[v_1, v_2] = [2.5\text{V}, 2.5\text{V}]$  and  $[v_1, v_2] = [2.5\text{V}, 1.5\text{V}]$ , respectively. The PRBS inputs were used in supplement to the steps to excite the high frequencies as adopted in Chapter 3.3.2. The outputs of the system were measured and are shown in Figure. 4.7. Then the same inputs are applied to the ODEs given by Equation (3.17) using a fourth-order Runge-Kutta numerical method. The error residuals between the actual and modelled responses,  $e_1$  and  $e_2$ , are shown in Figures 4.8(a) and (b) for two different operating point sets. The corresponding root mean-square (RMS) errors at operating point  $[v_1, v_2] = [2.5\text{V}, 2.5\text{V}]$  are 1.0641 and 1.4215 for  $h_1$  and  $h_2$ , respectively. For the operating point at  $[v_1, v_2] = [2.5\text{V}, 1.5\text{V}]$ , these are 3.0120 and 1.4919. It can be seen that the errors are large and dependant on the operating points, although the model is not linearised.

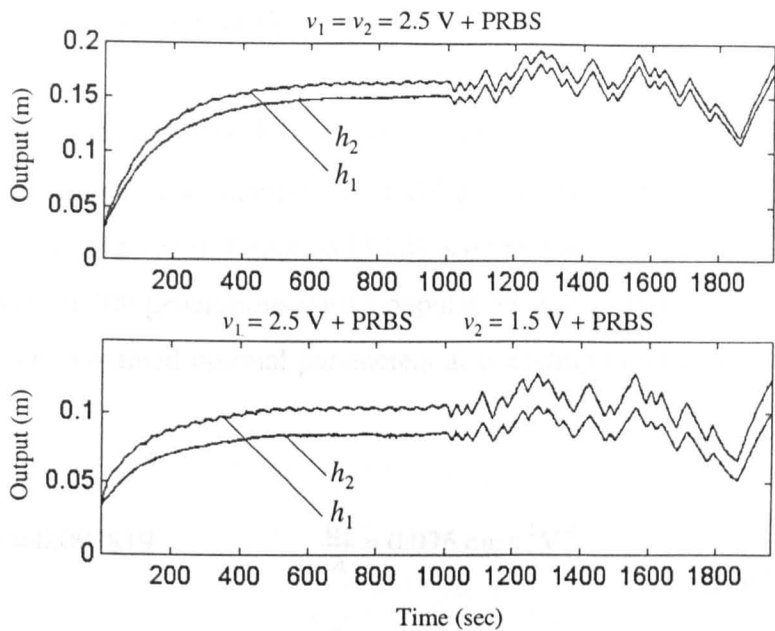


Figure 4.7 Noisy responses of the physical system to steps and PRBS exciting the interested frequency range.

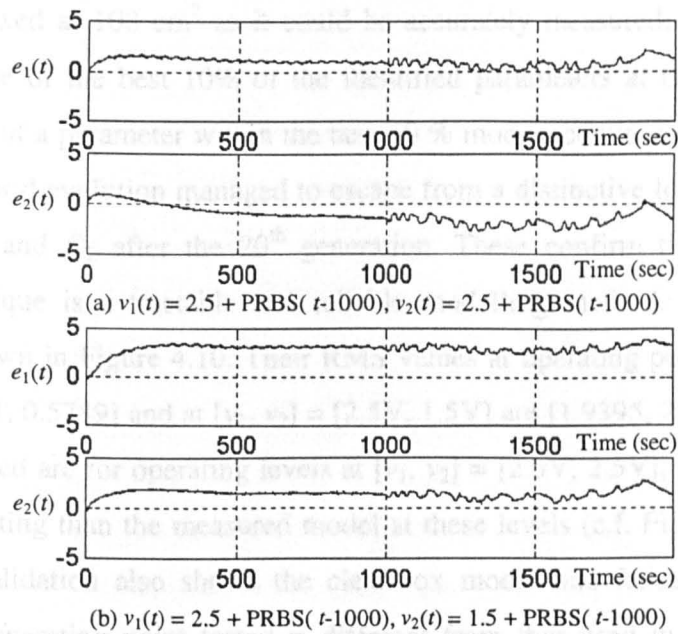


Figure 4.8 Error residuals (in cm) of the measured clear-box model at different operating levels.

### 4.3.2 Optimisation of the Clear-Box Model

To derive a more accurate clear-box model, evolutionary algorithm based identification technique has been applied to identify the model parameters in Equation (3.17). Here, the  $L_2$  norm cost function given in Equation (3.20) is employed. It took about 2 hours for the hybrid EA to run for 100 generations with a population size of 100 on a 200 MHz Pentium Pro processor. The obtained optimal parameters at operating point  $[v_1, v_2] = [2.5V, 2.5V]$  are given by

$$\begin{aligned} \frac{C_1 a_1}{A} &= 0.003819 & \frac{Q_1}{A} &= 0.076 \text{ cm s}^{-1} \text{V}^{-1} & H_3 &= 2.5 \text{ cm} \\ \frac{C_2 a_2}{A} &= 0.002002 & \frac{Q_2}{A} &= 0.051 \text{ cm s}^{-1} \text{V}^{-1} \end{aligned}$$

Note that  $A$  is fixed at  $100 \text{ cm}^2$  as it could be accurately measured. Figure 4.9 shows a convergence trace of the best 10% of the identified parameters at each generation. The candidate values of a parameter within the best 10 % models converged to a narrow range. Note that the hybrid evolution managed to escape from a distinctive local optimum caused mainly by  $C_2 a_2$  and  $Q_2$  after the 20<sup>th</sup> generation. These confirm that the evolutionary modelling technique is a tractable and reliable modelling method. The resulting error residuals are shown in Figure 4.10. Their RMS values at operating point  $[v_1, v_2] = [2.5V, 2.5V]$  are  $[0.5401, 0.5789]$  and at  $[v_1, v_2] = [2.5V, 1.5V]$  are  $[1.9395, 2.6033]$ , respectively. Since the data used are for operating levels at  $[v_1, v_2] = [2.5V, 2.5V]$ , the clear-box model offers a better fitting than the measured model at these levels (c.f. Figures. 4.8 and 4.10). However, the validation also shows the clear-box model had failed to offer a similar accuracy if the operating point tested is different from that used in obtaining the data. Although the clear-box model is not linearised, the validation shows that it is not suitable for modelling this practical system, as it ignores unmeasurable non-dominate nonlinearities.

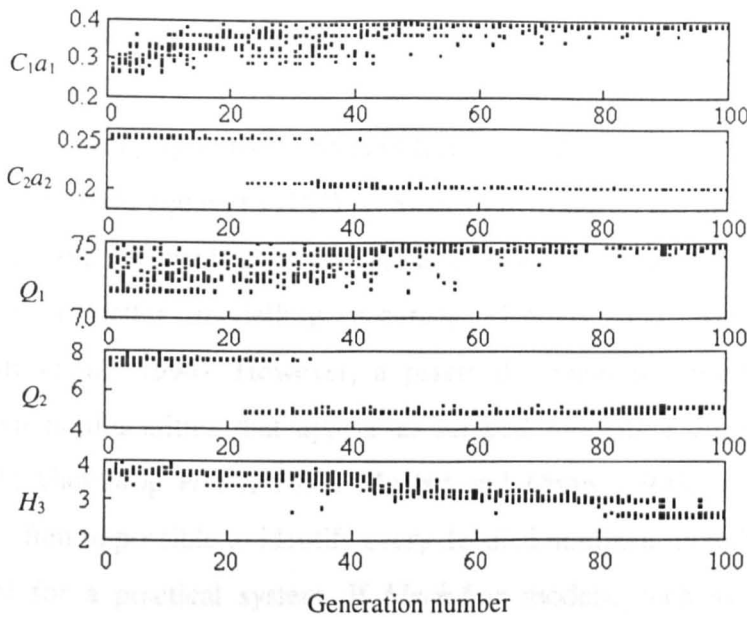


Figure 4.9 Convergence trace of the best 10% parameters in each generation

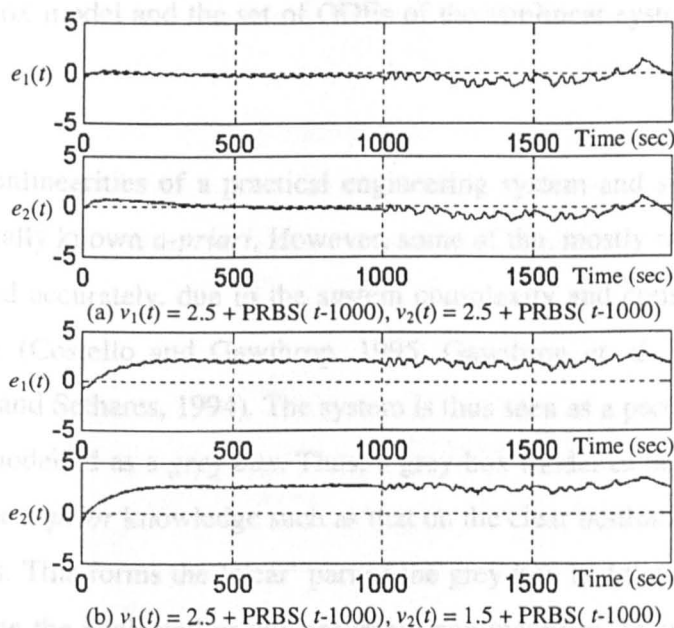


Figure 4.10 Error residuals (in cm) of the identified clear-box model at different operating levels

Such an identification problem is often a multi-modal optimisation problem in a multi-dimensional space. The optimisation task can hardly be performed by conventional techniques, which require a smooth search space or a differentiable error energy function. In addition, these conventional gradient-guided techniques may encounter difficulties due

#### 4.4 Evolutionary Grey-Box Modelling for Practical Systems

Many practical engineering systems involve nonlinearity, but are often described by a set of linear ordinary differential equations (ODEs). Since system ODEs are usually derived from the physical operating mechanism of the system, they should accommodate nonlinear descriptions for a better modelling accuracy (Kemna and Mellichamp, 1995; Vandemolengraft *et al.*, 1994). However, a practical system is complicated and may involve unknown nonlinearities that appear as lumped modelling errors (Costello and Gawthrop, 1995; Gawthrop *et al.*, 1993; Maclay and Dorey, 1993; Yao and Sethares, 1994). It is thus often impossible to identify every detailed nonlinearity to build an accurate *clear-box* model for a practical system. If *black-box* models, such as those based on nonlinear fuzzy, neural network or nonlinear auto-regressive moving average, are attempted instead, structural information on the system will be lost, because mappings between a black-box model and the set of ODEs of the nonlinear system are not bijective or equivalent.

The underlying nonlinearities of a practical engineering system and some of its physical parameters are usually known *a-priori*. However, some of the, mostly minor, nonlinearities cannot be modelled accurately, due to the system complexity and constraints on physical ability to measure (Costello and Gawthrop, 1995; Gawthrop *et al.*, 1993; Maclay and Dorey, 1993; Yao and Sethares, 1994). The system is thus seen as a partially-known system and may be best modelled as a *grey-box*. Thus, a grey-box model to be established should explicitly utilise the *a-priori* knowledge such as that on the clear nonlinear structure derived from physical laws. This forms the 'clear' part of the grey-box and the 'black' part will be used to approximate the neglected or unmeasurable nonlinearities. In such a grey-box, the system structure will not be replaced by artificial structures as seen in generic black-box approximators. This is in contrast to the conventional clear or black-box identification techniques.

Such an identification problem is often a multi-modal optimisation problem in a multi-dimensional space. The optimisation task can hardly be performed by conventional techniques, which require a smooth search space or a differentiable error energy function. In addition, these conventional gradient-guided techniques may encounter difficulties due

to noisy data or only offer “local optima” if the initial guess is inappropriate (Li *et al.*, 1997; Tan *et al.*, 1995). A simple and highly accurate novel grey-box modelling technique is thus developed in this Section, which is tractably enabled by the hybrid evolutionary algorithms developed in Chapter 2. The methodology is detailed with two nonlinear modelling examples in engineering applications.

4.4.1 Grey-Box Model of the Coupled Nonlinear System

The nonlinearities that have often been omitted in clear-box modelling of the coupled nonlinear system include non-dominant contributions by fluid tension and friction and include simplification from distributed pipes to lumped orifices. Other inaccuracies arise from manufacturing tolerance and measurement errors. Further, due to unknown and unattainable nonlinearities, empirical modelling shows that the ODE coefficients,  $C_1$  and  $C_2$ , are also operating-point dependent as shown in Figure 4.11. This dependency is in fact, a state dependency determined by the potential energies  $2g|h_1-h_2|$  and  $2g|h_2-H_3|$ . As shown in Figure 4.11, the two discharge coefficients of  $C_1$  and  $C_2$  at steady-state are non-linearly dependent on the potential energies of  $2g|h_1 - h_2|$  and  $2g|h_2 - H_3|$ , respectively.

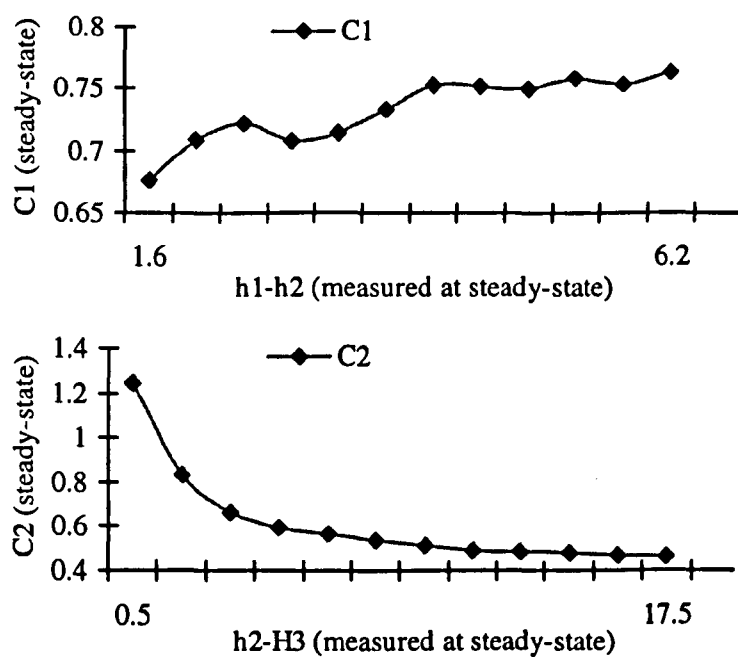


Figure 4.11 Operating level dependent ‘coefficients’ in a nonlinear clear-box model



For engineering applications like the twin-tank system, it is often impossible to identify every detailed nonlinearity to obtain an accurate clear-box model, but it is often desired to have a focused physical model structure like that in a clear-box model in Equation (3.17). For this, a novel type of grey-box model is proposed here. This model builds from the clear-box model as the global structure and incorporates local black-boxes to model the operating-point sensitive coefficients that are unable to model by a clear-box model due to neglected nonlinearities. Such black-boxes can be power series polynomial, fuzzy logic, neural network or other types of basis function based generic function approximators. Since Padé approximation in the form of a regressive function is accurate and efficient, it is recommended here to play the black-box role in the combined grey-box model.

For Equation (3.17), the two sensitive “coefficients” are now modelled by the black-box:

$$\frac{C_1 a_1}{A} = \frac{c_{10} + c_{11}|h_1 - h_2| + c_{12}|h_1 - h_2|^2 + \dots}{A(1 + d_{11}|h_1 - h_2| + d_{12}|h_1 - h_2|^2 + \dots)} \quad (4.27)$$

and

$$\frac{C_2 a_2}{A} = \frac{c_{20} + c_{21}|h_2 - H_3| + c_{22}|h_2 - H_3|^2 + \dots}{A(1 + d_{21}|h_2 - H_3| + d_{22}|h_2 - H_3|^2 + \dots)} \quad (4.28)$$

where  $c_{ij}$  and  $d_{ij}$  are absolute constants. For the resultant grey-box model that combines Equations (3.17), (4.27) and (4.28), the modelling task is to find  $H_3$ ,  $Q_1/A$ ,  $Q_2/A$ ,  $c_{ij}$  and  $d_{ij}$  such that the  $L_2$  norm cost function given in Equation (3.20) is minimised.

Since  $H_3$  in Equation (3.17) cannot be linearly parameterised or separated from  $C_2 a_2/A$  and the black-boxes need to be identified simultaneously within the nonlinear clear-box model structure, conventional parameter estimation techniques can hardly be applied here. However, in an evolutionary modelling process, there is no need for linearly separable parameterisation as in conventional identification methods (Kemna and Mellichamp, 1995). Candidate models in evolution can combine physical and empirical models (Tan *et al.*, 1997) and the evolution can start from empirical ones even when they are clear-box

models initially. Optimal orders of the Padé regressions can also be found in the same process if a structural encoding technique is used (Li *et al.*, 1997).

The parameters of the grey-box model resulting from the evolution are shown in Table 4.3. More importantly, the grey-box also reveals the trends of  $C_1a_1$  and  $C_2a_2$  as the potential energies change, as shown in Figure 4.12. The RMS grey-box modelling errors are given in the last rows of Table 4.4 and Table 4.5 for the two different operating levels. For a clear comparison, these values obtained from the measured and the clear-box models are also shown in Tables 4.4 and 4.5. The modelling error residuals resulting from the grey-box are shown in Figure 4.13. It can be seen that the model not only provides significantly improved identification quality but also is robust to operating levels.

Table 4.3 The evolved grey-box model parameters

Padé efficients	For $C_1a_1$	For $C_2a_2$
$c_{i0}$	0.03734	0.4004
$c_{i1}$	0.3789	0.0293
$c_{i2}$	0.27728	0.01516
$d_{i1}$	0.2583	0.0811
$d_{i2}$	0.8148	0.072
$H_3 = 3.1$	$Q_1 = 7.2$	$Q_2 = 7.5$

Table 4.4 The RMS errors of the measured clear-box, identified clear-box and identified grey-box models

Mean excitation voltages	$v_1 = v_2 = 2.5V + PRBS$	
RMS errors (in cm)	Tank 1: $e_1$	Tank 2: $e_2$
Measured clear-box model	1.0641	1.4215
Clear-box identification	0.5401	0.5789
Grey-box identification	0.2906	0.2854

Table 4.5 The RMS errors at different operating-point

Mean excitation voltages	$v_1 = 2.5V + PRBS;$ $v_2 = 1.5V + PRBS$	
RMS errors (in cm)	Tank 1: $e_1$	Tank 2: $e_2$
Measured clear-box model	3.0120	1.4929
Clear-box identification	1.9395	2.6033
Grey-box identification	0.5511	0.2816

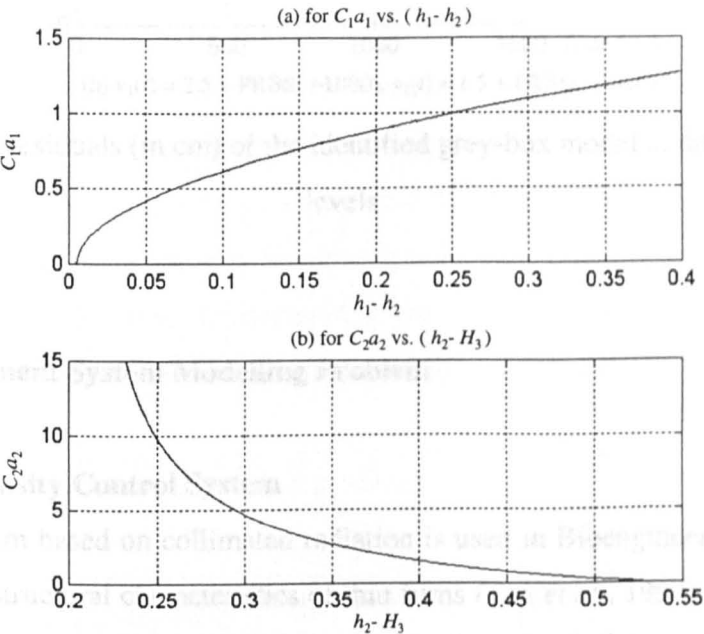


Figure 4.12 Modelling the varying ‘coefficients’ of the clear structure in the grey-box by black-boxes

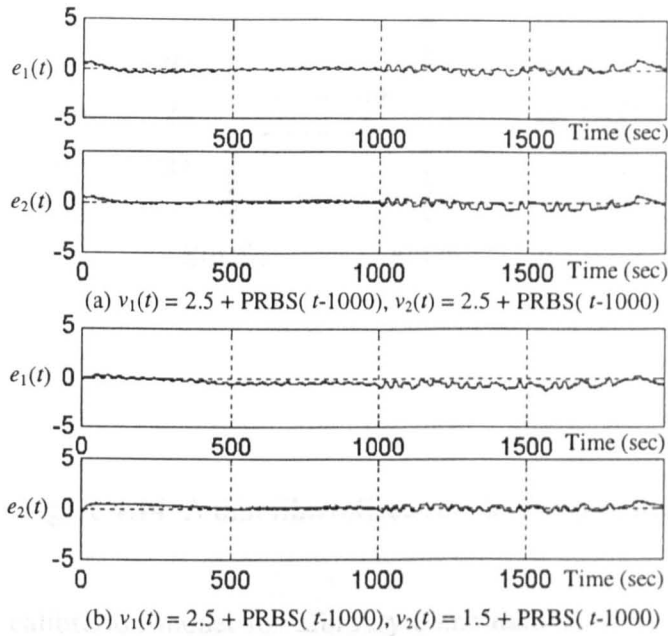


Figure 4.13 Error residuals (in cm) of the identified grey-box model at different operating levels

4.4.2 A Measurement System Modelling Problem

The Neutron Intensity Control System

A reflectivity system based on collimated radiation is used in Bioengineering to determine the thickness and structural characteristics of thin films (Tan *et al.*, 1997). A neutron beam hitting the film sample is reflected to the monitoring detector as shown in Figure 4.14. To get a complete description of the reflectivity profile, the sample is rotated so that the incident beam strikes it at different angles. Because the rotation will vary the intensity of the reflected beam, it is necessary to control the size of the neutron beam hitting the reflecting sample by means of two slits  $S_1$  and  $S_2$  when the sample is rotated. To analyse the thin film sample using its reflectivity, the intensity of the beam hitting the sample,  $A_d$ , must be known so that calibrations can be made. This is difficult for arbitrary sized combinations of  $S_1$  and  $S_2$  values. Thus, an inverse model needs to be generalised for automatic calibrations.

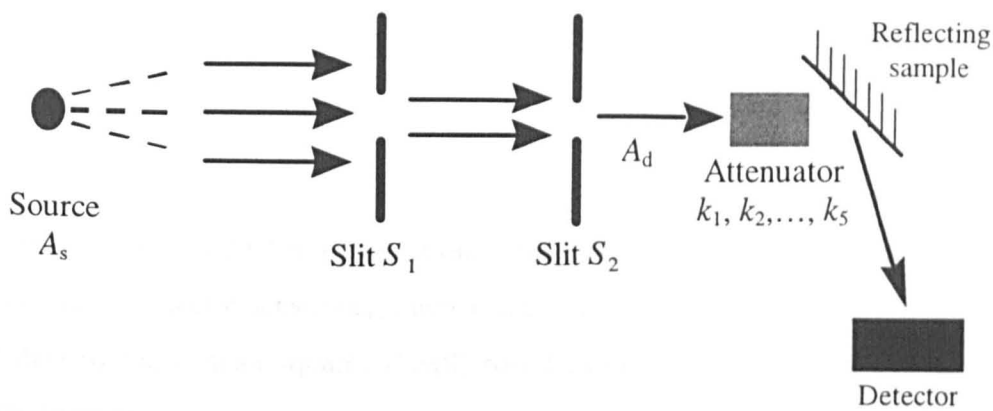


Figure 4.14 A thin film reflectivity detecting system

To generalise the calibration model for arbitrary combinations of  $S_1$  and  $S_2$ , the intensity responses to more than 60 pre-determined combinations of  $S_1$  and  $S_2$  are obtained as training data. To measure the data accurately within the sensitivity range of the detector while the slit sizes are changed, the manufacturer of the system recommends to add an attenuator after Slit 2. However, the attenuation can only be varied with the product of the two slit sizes by 5 discretely decreasing values, as shown in Table 4.6.

Table 4.6 Discretely decreasing attenuating factors

$S_1 S_2$	Attenuating factors $k_i(S_1 S_2)$	Estimated values
$[0, 0.25)$	$k_1 (> k_2)$	3900
$[0.25, 2.0)$	$k_2 (> k_3)$	739.8
$[2.0, 9.0)$	$k_3 (> k_4)$	151.76
$[9.0, 44.0)$	$k_4 (> k_5)$	29.16
$[44.0, 207.0)$	$k_5 (> 0)$	5.46

By physical principles, the ‘gain’ from the input beam intensity,  $A_s$ , to the output beam intensity,  $A_d$ , would be proportional to the product of the cascaded slit sizes:

$$\hat{I} = \frac{A_d}{A_s} = c \, k_i(S_1 S_2) \, f(S_1, S_2) \tag{4.29}$$

where

$$f(S_1, S_2) = S_1 S_2 \quad (4.30)$$

In Equation (4.29),  $c = 7.63$  is a proportional factor of the gain, depending upon the source intensity and the detector sensitivity. Such a clear-box model was identified using three sets of I/O data by least mean squares (LMS) based estimation (Grace, 1992) with the cost function given by:

$$J = \sum_{m=1}^3 \sum_n \left( I - \hat{I} \right)^2 \quad (4.31)$$

where  $I$  represents the measured data,  $n \geq 60$  represents the number of measurements made in one data set. One such data set is depicted in Fig. 4.15(a). The reason to use three data sets was to average down the measurement errors. The estimated attenuation factors with  $c = 7.63$  are given in the third column of Table 4.6. Unfortunately, the validation exercise carried out concluded that the identified model failed to provide a good fitting, with an RMS error of 365.1. The performance of this method also depended critically upon the initial guess.

### Evolutionary Grey-Box Modelling

The neglected diffraction effects and other unknown factors in the clear-box modelling may, however, be accounted for by a Padé approximation based black-box as discussed in the previous section. Based on this, a grey-box model may be established. The clear-box model structure of Equation (4.29) will be preserved in the formation of the grey-box here, whilst Equation (4.30) will be replaced by a Padé black-box. For example,  $S_1$  in Equation (4.30) may be expanded to  $(a_0 + a_1 S_1 + a_2 S_2)$  in order to include diffraction and other neglected and difficult-to-model effects. A general form of the black-box can thus be easily written in the form:

$$f(S_1, S_2) = \frac{(a_0 + a_1 S_1 + a_2 S_2)(b_0 + b_1 S_1 + b_2 S_2)}{c_0 + c_1 S_1 + c_2 S_2} \quad (4.32)$$

where  $a_i$ ,  $b_i$ , and  $c_i$  are the model coefficients to be determined. Here, flexibility exists such that  $S_1$  and  $S_2$  may be allowed to have asymmetric contributions. A grey-box model candidate is obtained by combining Equation (4.29) and Equation (4.32). Note that the inclusion of the denominator may be unnecessary but may be useful for including some insignificant general nonlinearities. The same three data sets of  $I$  against  $S_1$  and  $S_2$  are used to derive the grey-box coefficients of the neutron intensity control system. The aim is to provide the best set of  $a_i$ ,  $b_i$ ,  $c_i$  in Equation (4.32) and  $ck_i$  in Equation (4.29) such that the amalgamated modelling error for all the three data sets shown in Equation (4.31) is minimal.

The same evolutionary grey-box modelling technique is applied here to determine the grey-box model. The resulting parameters and RMS error at the end of the evolution are shown in column 2 of Table 4.7. The responses of the grey-box model are shown in Figure 4.15(b) — (d). Obviously, the identified model performed a good fitting to all the three data sets with small relative error residuals. The evolution correctly yielded zero coefficients of  $a_0$  and  $b_0$ , as  $S_1 = S_2 = 0$  would result in a zero brightness count. Further inspection at  $c_1$ ,  $c_2$ ,  $a_1$   $b_1 = 0.008068$  and  $a_2$   $b_2 = 0.005025$  reveals that the fitted Equation (4.32) is not completely symmetrical and should have included the neglected non-reciprocal effects such as by diffraction. Those small coefficients within  $[0, 0.016]$  indicate that Equation (4.32) is indeed close to the dominant structure of Equation (4.30).

Equation (4.30) may also be enhanced by a more ‘black’ Padé model as written in the expanded form:

$$f(S_1, S_2) = \frac{a_0 + a_1 S_1 + a_2 S_2 + b_0 S_1 S_2 + b_1 S_1^2 + b_2 S_2^2}{1 + C_1 S_1 + C_2 S_2} \quad (4.33)$$

Here, the constant term in the denominator is fixed as 1, since its value is not expected to be zero. The evolved model coefficients and RMS error are given in column 3 of Table 4.7. Again, the evolution recommended no first-order numerator terms and small denominator corrections. Obviously, the obtained results are consistent with the black-box representation given by Equation (4.32), which confirms the robustness and reliability of the proposed methodology.

To compare the results, the same hybrid evolutionary algorithm is also applied to identify the clear-box model of Equation (4.29) and Equation (4.30) derived from the physical principles. Results of this clear-box modelling are given in column 4 of Table 4.7. The RMS error is larger than those of the grey-boxes, since the symmetric clear-box model of  $S_1$  and  $S_2$  in Equation (4.29) and Equation (4.30) neglected non-dominant's characters. For convenience of comparison, the clear-box model coefficients estimated using a LMS method are re-listed in the fifth column of Table 4.7. It can be seen that the LMS method performed less satisfactory than the hybrid EA.

Table 4.7 The evolved grey-box model parameters

Parameters	Grey-box Equations (4.29) and (4.32)	Grey-box Equations (4.29) and (4.33)	Clear-box	
			EA	LMS
$a_0$	0	0		
$a_1$	0.8068	0		
$a_2$	0.01	0		
$b_0$	0	0.2853		
$b_1$	0.01	0.0137		
$b_2$	0.5025	0.003		
$c_0$	1 (normalised)	1 (fixed)		
$c_1$	0.0113	0.016		
$c_2$	0.0119	0.012		
$ck_1$	67026.6	88638.6	28628	29757
$ck_2$	14464.8	19219.2	6066	5645
$ck_3$	2852.5	3818.8	1157	1158
$ck_4$	570.5	779.8	215	222.5
$ck_5$	100	139	34.4	42
RMS error	115	113	150	365



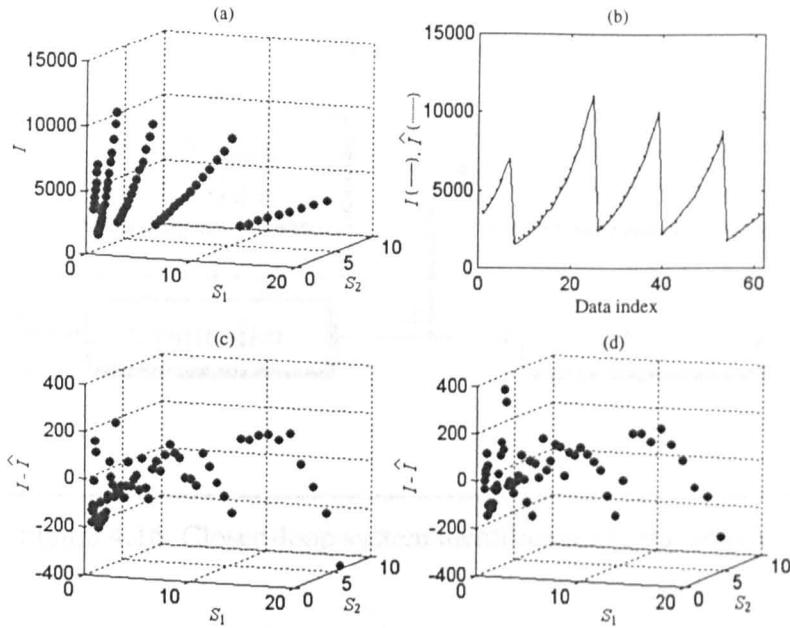


Figure 4.15 (a) Measured gain in data set 1; (b)  $I$  {—} and  $\hat{I}$  {....} fitted by the grey-box of data set 1; (c), (d) Error residuals for fitted data set 2 and 3, respectively

#### 4.5 Evolution Based Closed-Loop System Identification

In control system practice, an engineer often requires an accurate model of the system to be controlled. Using evolutionary techniques reported earlier in the thesis, the model can be identified from the open-loop plant I/O data and can be further refined off-line. The same EA can then be applied to obtain the best controller, which will be studied later in Chapters 5 and 6. When on-line, however, the designed controller may not perform the best due to uncertainties of the model, changes in process dynamics or disturbances that may occur in operations. For unstable systems, open-loop data of plant response may be unsuitable or unavailable. Under these circumstances, closed-loop system identification is often desired, where parameters need to be identified or re-adjusted based on observed closed-loop control signal and process output as shown in Figure 4.16. Note that the closed-loop system identification carried out in this section is from off-line data. Compared with open-loop data, the plant stimuli are limited by the control signal and thus the identification task would be more difficult.

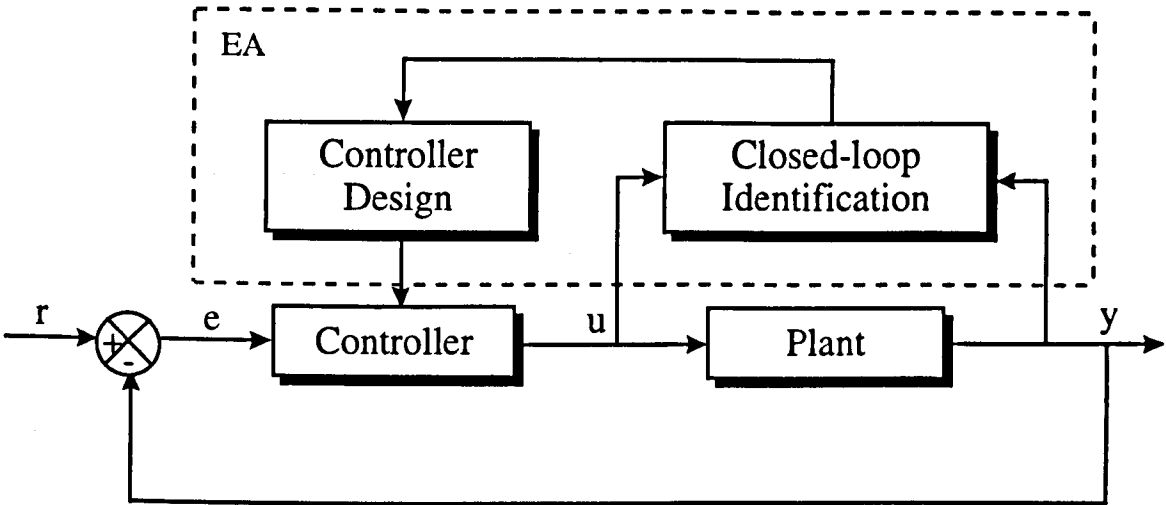


Figure 4.16 Closed-loop system identification using an EA

4.5.1 Closed-Loop Identification for a Linear System

The DC servo-system studied in Section 4.2.1.2 and later in Section 5.4.1 is experimented here. As will be revealed in Section 5.4.1, an EA designed controller based on the clear-box model described by Equation (4.4) is given by  $H(s) = 77.68 \frac{0.01s^3 + 0.31s^2 + 1.16s + 1}{s^3 + 23.82s^2 + 29.23s + 0}$ . This controller has been implemented for the physical servo-system. As illustrated in Figure 4.16, the task here is to use the hybrid EA detailed in Chapter 2 to further adjust or fine-tune the clear-box model parameters from the implemented control signal and output response. The control signal shown in Figure 4.17 is used as the plant stimuli. To make use of available knowledge of the open-loop identified parameters obtained in Section 4.2.1.2, the searching space in the EA are restricted within the ranges of  $\pm 50\%$  of their values shown in Table 4.2. The  $L_2$  norm given in Equation (4.3) is used here as the cost function. The hybrid EA with a population size of 20 has been run for 20 generations. It took about 40 seconds for the EA to complete on a Pentium Pro 200 MHz processor. This speed has shown good potential of EAs for on-line identification and on-line controller design for adaptive control. The final EA evolved model parameters are shown in the second column of Table 4.8. Figure 4.18 shows the response of the closed-loop identified clear-box model against the actual response of the physical system. It can be seen that the identification result is satisfactory, with a reasonably accurate fitting to the physical system.

Table 4.8 Model parameters obtained from the evolution based closed-loop system identification

Linear DC servo-system		Nonlinear coupled liquid-level system	
Independent Parameters in Equation (4.4)	EA identified parameters	Independent Parameters in Equation (3.17)	EA identified parameters
$\frac{JR + LB}{LJ}$	6.97	$\frac{C_1a_1}{A}$	0.0045
$\frac{RB}{LJ}$	7.47	$\frac{C_2a_2}{A}$	0.0026
$\frac{K_T}{LJ}$	4.61	$\frac{Q_1}{A}$	0.13
$T$	0.06	$\frac{Q_2}{A}$	0.04
		$H_3$	3
RMS error	0.0314	RMS error for Tank 1	0.515
		RMS error for Tank 2	0.367

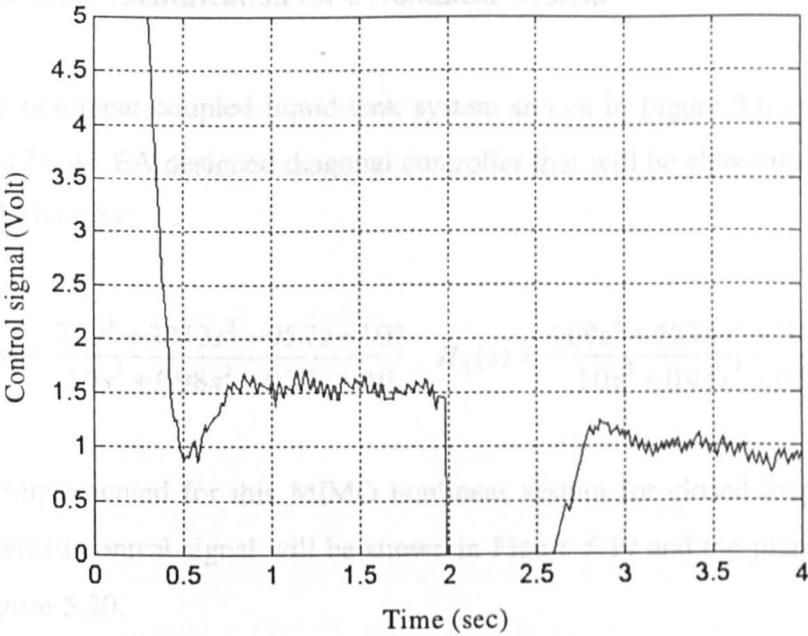


Figure 4.17 Control signal used for the closed-loop identification

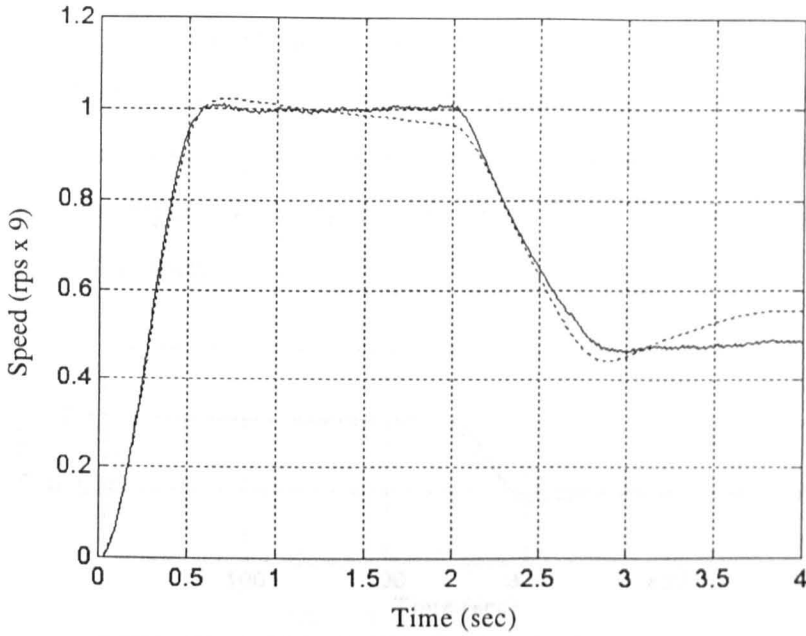


Figure 4.18 Responses of the DC servo-system{ — }and the EA identified clear-box model from closed-loop data { - - }

4.5.2 Closed-Loop Identification for a Nonlinear System

Consider the nonlinear coupled liquid-tank system shown in Figure 3.6 and described by Equation (3.17). An EA designed diagonal controller that will be elaborated in Section 5.5 has the TFM given by:

$$H_1(s) = \frac{7.9s^3 + 30.13s^2 + 95.7s + 1.02}{1.0s^3 + 0.98s^2 + 0.73s + 0.0}, H_2(s) = \frac{4.69s^3 + 55.76s^2 + 57.56s + 0.86}{1.0s^3 + 0.46s^2 + 0.38s + 0.0}$$

It has been implemented for this MIMO nonlinear system for closed-loop identification. The implemented control signal will be shown in Figure 5.19 and the plant output will be shown in Figure 5.20.

A nonlinear clear-box model closed-loop identification is considered here. The  $L_2$  norm cost function given in Equation (3.20) is used in the EA. The searching space in the EA are restricted within the ranges of  $\pm 50\%$  of the open-loop identified parameters given in

Section 4.3.2. It took less than 1 minute for the hybrid EA to run for 20 generations with a population size of 20 on a Pentium Pro 200 MHz processor. The EA recommended model parameters are shown in column 4 of Table 4.8. The response of the EA identified model is shown by the dotted line in Figure 4.19. To compare, the plant output shown in Figure 5.20 is also plotted in Figure 4.19, by a solid line. These indicate again a satisfactory identification performance.

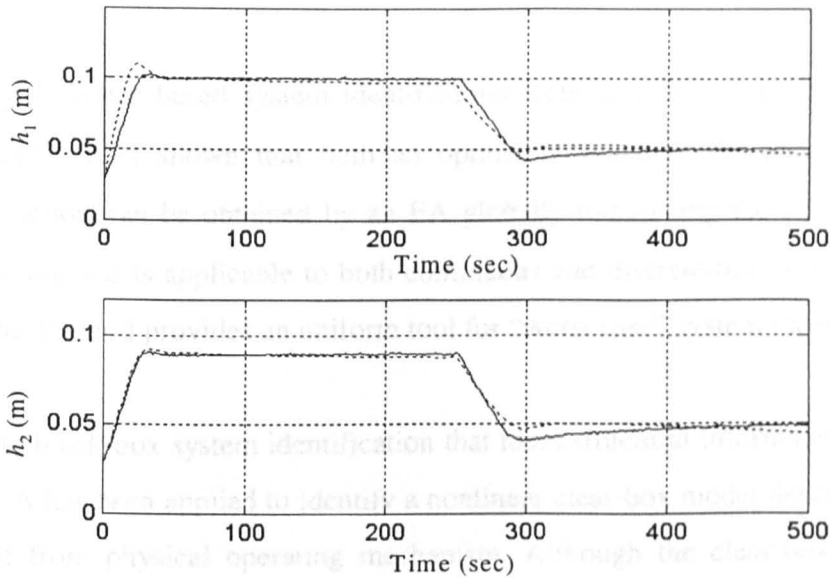


Figure 4.19 Responses of the physical nonlinear system { — } and EA identified clear-box model from closed-loop data { - - - }

#### 4.6 Summary

This chapter has developed a  $L_2$  and  $L_\infty$  norm based system identification technique using the hybrid evolutionary algorithms detailed in Chapter 2. Although  $L_2$  and  $L_\infty$  norms are considered for the identification performance measure here, other measures such as  $L_1$  norm can also be employed if desired. The choice of the error measure should, however, depend on the purpose that the identified model is to be served.

This hybrid EA technique has been applied to identify the parameters of a system described by a black-box based on ARMAX model with the presence of white noise. The technique

has also successfully identified a DC servo-system represented by an ODE based clear-box model. Step input based identification has been developed, since step response data are common and readily available for most engineering systems. This technique could be used to determine the transient and steady-state behaviour of a system rapidly and could be applied to on-line system identification. Identification results obtained from the evolutionary technique is shown to be more accurate compared with the conventional least mean-squares method.

Evolutionary  $L_\infty$  norm based system identification technique for robust control has also been developed. It is shown that both an optimal nominal model and an uncertainty bounding function can be obtained by an EA globally minimising the  $L_\infty$  norm costs. In addition, the method is applicable to both continuous and discrete-time systems. Example show that this method provides an uniform tool for “worst case” system identification.

In contrast to black-box system identification that loses structural information of a system, the hybrid EA has been applied to identify a nonlinear clear-box model described by ODEs that derived from physical operating mechanism. Although the clear-box identification results for a coupled twin-tank nonlinear system are satisfactory, it is observed that the identification errors are largely dependent on the operating points, although the model is not linearised. This clear-box model is thus not suitable for modelling this nonlinear system.

A novel evolution based grey-box modelling technique has been developed. The technique makes the best use of *a-priori* knowledge on the clear-box global structure of a physical system, whilst incorporates accurate black-boxes for unmeasurable local nonlinearities. Such grey-box models can hardly be established by conventional parameter estimation techniques but can be easily achieved through generational convergence offered by an evolutionary algorithm and through individual learning by Boltzmann selection. The evolution can start from empirical models, making best use of existing knowledge on a practical system. The examples of a hydraulic nonlinear system and a neutron intensity control system have shown the feasibility and power of the proposed modelling methodology. The grey-box modelling and hybrid evolution techniques could be easily

applied to many linear or nonlinear industrial plants. It is expected that this type of grey-box models will accommodate many practical systems.

Evolution based closed-loop system identification has also been studied. This identification technique is validated by clear-box models. Experimentation results of both linear and nonlinear physical systems have shown satisfactory identification performances. In addition, the identification could be achieved in a relatively short processing time, which shows good potential of EAs for on-line adaptation.

## Chapter 5

# UNIFICATION AND AUTOMATION OF LINEAR CONTROLLER DESIGN FOR LINEAR AND NONLINEAR PLANTS

## 5.1 The Problem of Design

As addressed in the Introduction in Chapter 1, the design process of a linear control system usually involves the optimisation of a number of parameters of the controller in order to obtain one or more optimal or sub-optimal designs that meet a number of performance requirements under certain practical constraints. This *design problem* is thus equivalent to a multi-modal optimisation problem in an, almost certainly, multi-dimensional space. If the *objective function* (or *cost function*) is differentiable under practical constraints in the multi-dimensional space, the design problem may be solved easily by setting its *vector derivative* to zero. Finding the parameter sets that result in zero first-order derivatives and that satisfy the second-order derivative conditions would reveal all *local optima*. Then comparing the performance values of all the local optima, together with those of all boundary parameter sets, would lead to the *global supremum*. The corresponding parameter set would thus represent the best controller meeting the design specifications.

However, the design objective may not be differentiable and can be discontinuous in practice. This is mainly due to practical design specification terms, such as minimum order and logic terms, and constraints of the physical system, such as nonlinearities, actuator saturation, transportation delay and noisy data. This means a design problem is hard to solve. On contrast, a *simulation problem* of a practical control system can now be solved easily. Existing computer-aided control system design (CACSD) packages are just the tools for this. Note that most of these design packages only provide a passive simulation tool for control engineers with few direct or automated design facilities available (Li *et al.*, 1995a).

This Chapter attempts to solve these problems in an uniform way. The following section begins with discussions on design difficulties encountered in conventional numerical



methods. Then associated problem-classification and a possible automation technique are studied. Practically realisable design automation methodology and limitations using evolutionary algorithms are addressed in Section 5.2. In Section 5.3, this methodology is to be applied to the design unification in the form of uniform LTI control (ULTIC) systems, after frequency-domain and time-domain formulations of the unification are discussed. ULTIC system design examples for linear and nonlinear plants will be presented in Section 5.4. A MIMO ULTIC system design is studied in Section 5.5. To make the design more visualisable, a multi-objective evolutionary algorithm will be incorporated in ULTIC system design in Section 5.6. A parallel EA will be shown in Section 5.7 to assess the effectiveness of parallelism in the ULTIC system design. Summary will be drawn in Section 5.8.

### 5.1.1 Design Specification and Problem Formation

In control system design practice, the structure of a controller is usually determined by the control scheme or control law that the design engineer chooses to use. Thus most design tasks are to optimise the parameters of the controller so that it best meets the design objectives or customer specifications. This means that a parameter set of the controller represents a design candidate of the system.

**DEFINITION 1** In the context of design, a *candidate control system*,  $P_i$ , can be defined by a uniform vector representation given by:

$$P_i = \{p_1, \dots, p_n\} \in \mathbb{R}^n \tag{5.1}$$

where  $i$  stands for the  $i^{\text{th}}$  design candidate,  $n$  the number of parameters required by the control law,  $p_j \in \mathbb{R}$  the  $j^{\text{th}}$  parameter of the  $i^{\text{th}}$  design candidate with  $j \in \{1, \dots, n\}$ , and  $\mathbb{R}^n$  the  $n$ -dimensional real Euclidean space. Here the design space is assumed to be countable, since a countable number of design choices are usually considered in engineering practice (Li, 1996).

**DEFINITION 2** The solution space of a control system design problem can be defined by

$$\mathbf{S} = \left\{ P_i, \forall i \mid p_j \in \mathbf{R} \text{ and } j \in \{1, \dots, n\} \right\} \subseteq \mathbf{R}^n \quad (5.2)$$

**DEFINITION 3** The fitness of a control system design,  $P_i$ , can be defined by a function  $f(P_i): \mathbf{R}^n \rightarrow \mathbf{R}^+$  which represents the performance index of the control system with respect to the design requirements or specifications, where  $\mathbf{R}^+$  is the non-negative real space.

For control engineering applications, fitness function should reflect the performance requirements and customer specifications, such as:

*Spec. 1:* A good relative stability of the closed-loop system (e.g. gain-margin  $\in [4 \text{ dB}, 6 \text{ dB}]$ , phase-margin  $\in [40^\circ, 60^\circ]$  or 3 dB bandwidth  $\leq 10 \text{ Hz}$  etc.);

*Spec. 2:* An excellent transient response in terms of small rise-time, settling-time, overshoots and undershoots (e.g. overshoots  $< 10 \%$ );

*Spec. 3:* An excellent steady-state accuracy in terms of small steady-state errors (e.g. s.s.e.  $< 3 \%$ );

*Spec. 4:* Robustness in terms of disturbance rejection; and

*Spec. 5:* Robustness in terms of parameter uncertainty or sensitivity attenuation.

**DEFINITION 4** A control system design problem can be defined as the problem of finding a design given by:

$$P_o = \left\{ P_o \in \mathbf{S} \mid f(P_o) = \sup_{\forall P_i \in \mathbf{S}} f(P_i) \right\} \quad (5.3)$$

It was highlighted in Chapter 1 that many practical control system design problems are *unsolvable problems* in the analytical domain, while practical control system simulation problems are *solvable problems* in the numerical domain (Li *et al.*, 1995a; Li, 1996; Sedgewick, 1988). The following question is, however, left unanswered:

*Are practical control system design problems solvable in the numerical domain?*

5.1.2 Problem Classification

Before answering the above question, it is desirable to review the problem-classification used in computer science and algorithm engineering (Sedgewick, 1988). This is depicted in Figure 5.1 (Li *et al.*, 1995a). The clear area represents the set of unsolvable problems and the shaded areas represent solvable problems.

The solvable problems are further divided into three categories as follow:

- $P = \{\text{Problems that can be solved by a deterministic algorithm in polynomial time}\};$
- $NP\text{-}COMPLETE = \{\text{Problems that cannot be solved by any deterministic algorithms in polynomial time but can be solved by a non-deterministic algorithm in polynomial time}\};$  and
- $\overline{NP} = \{\text{Problems that cannot be solved in polynomial time but can be solved otherwise}\}$

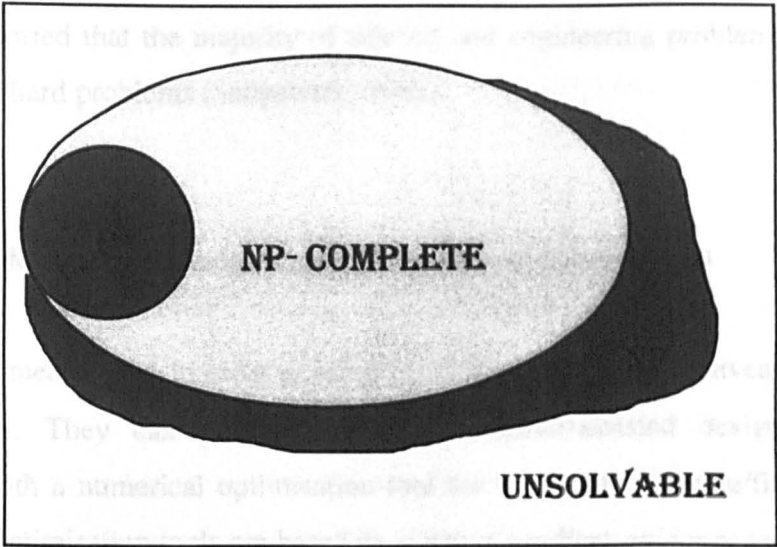


Figure 5.1 Classification of numerical problems

Not directly shown in Figure 5.1, the following definitions also exist (Li, 1996):

$$P \cup NP\text{-COMPLETE} = NP$$

= {Problems that can be solved in polynomial time}; and

$$\overline{NP} \cup NP\text{-COMPLETE} = NP\text{-HARD}$$

= {Problems that cannot be solved by any deterministic algorithms in polynomial time}

= {Problems that are at least as hard as an *NP* problem}.

It should be noted that the hypotheses:

$$NP\text{-COMPLETE} = \emptyset;$$

$$P \supseteq NP; \text{ or}$$

$$P = NP$$

have never been proven untrue and still remain a mystery in computer science today, despite a great deal of research efforts have been made during the past several decades (Sedgewick, 1988). The above classification has, however, encountered no controversial cases so far and is widely accepted in computer science and algorithm engineering (Li, 1996). It is reported that the majority of science and engineering problems belong to the category of *NP*-hard problems (Sedgewick, 1988).

### 5.1.3 Unsolvability of Practical Design Problem by Conventional Means

The numerical means used in most existing CACSD packages are conventional calculus-based methods. They can perform well in computer-assisted design, if they are incorporated with a numerical optimisation tool for quadratic objective/fitness functions. Conventional optimisation tools are based on *a-priori gradient-guidance* techniques. Some control system design problems could be transformed by these techniques to a *P*-problem, resulting in a polynomial design time. Such scenario does hardly, however, exist in

practical systems. In dealing with practical design problems, there exist the following drawbacks of conventional techniques:

- (1) *Existence Problem*: Gradient guidance passively adjusts  $P_i$  using  $\nabla f(P_i)$  or well-defined smooth slopes of the objectives (Goldberg, 1989);
- (2) *Multi-modal Problem*: The sequentially guiding process usually leads to a local optimum (Goldberg, 1989) and is difficult to evaluate  $P_i$  at the boundary of  $S$ , although optimisation at parallel points may overcome this problem to a certain extent;
- (3) *Practical Problem*: The method is impossible to work with hard constraint conditions (Michalewicz, 1994) or non-numerical (logic) type of constraints. In addition, it does not work properly in the noisy search space (Goldberg, 1989) in practical applications;

Using a CACSD package based on these techniques for design, a design engineer usually needs to solve these problems by heuristic simulations. He/She has first to input certain *a-priori* controller parameters, such as those obtained from some preliminary analysis, and should then undertake simulations and evaluations using the package. If the simulated performance of the “designed” control system does not meet the specification, the designer would modify the values of the parameters randomly or by his/her real-time gained experience. The engineer would then run the simulations repeatedly until a “satisfactory” design emerges. Clearly, such a design process is neither automated nor easily carried out, since mutual interactions among parameters are hard to predict (*multi-dimensional problem*).

A possible alternative to this manual approach is to incorporate an *approximation algorithm* (Sedgewick, 1988) or a *random-walk* technique (Goldberg, 1989) in the optimisation process to achieve a computerised design. Convergence towards the correct directions or towards the global optimum is not, however, guaranteed (Michalewicz, 1994). Using these techniques, it is easy to answer the question:

*Is there a design such that  $f(P_i) < f_0$  ?*

but difficult to answer:

*Is there a design such that  $f(P_i) \geq f_0$  ?*

where  $f_0 \in \mathbf{R}^+$ . Thus, the resulting “satisfactory” design may not offer the best or near-best performance.

In addition, a modern paradigm of CACSD should also meet the *open environment* and other *design challenges* as listed below (Li *et al.*, 1995a; Barker, 1995)

- (1) Complexity of practical systems;
- (2) Required high quality and accuracy of design;
- (3) Speed of design;
- (4) Competition with available design tools (in terms of ease of use, for example); and
- (5) Robustness, reliability and safety arising from the design.

#### 5.1.4 Solvable Problem by Non-NP Numerical Means

Since the analysis problem is solvable and encounters no difficulties as those highlighted by the conventional drawbacks in the numerical domain, one approach achieve a solvable and possibly automated design could be to *exhaustively* evaluate in  $\mathbf{S}$  all the possible design choices  $P_i, \forall i$ . To illustrate this method, let  $n = 10$  and suppose in  $\mathbf{S}$  each parameter has 8 possible values or choices. Then there are a total of:

$$\max (i) = 8^{10} \tag{5.4}$$

permutations of design choices. Every candidate design here could be encoded by a string of 10 integers. By *enumerating* all based-8 integers of the string one by one could span the entire quantised design space  $\mathbf{S}$ . Now suppose that each evaluation by numerical simulation takes 0.001 second on an extremely fast computer. Then the entire design process would take  $0.001 \text{ second} \times 8^{10} \approx 13 \text{ days}$  to complete. This may be too time-consuming to accept in practice.

Although such a search scheme does transform an unsolvable problem to a solvable *non-NP* problem, the search time is in the order of  $O(p^n)$ , where  $p$  is the quantisation dimension of the parameters. Thus, exhaustive/enumerative schemes will break down on problems of “moderate” dimensionality and complexity (Bellman, 1957; Goldberg, 1989). Although there exist some specialised, or problem-dependent, numerical schemes which work more efficiently than the exhaustive search, they are confined to a narrow problem domain.

The exponential search time required by an exhaustive search mechanism could, however, be largely reduced, if *a-priori* experience of the designer could be incorporated in the search and the interim results of the evaluations could be used to guide the search intelligently. The following section presents techniques that achieve these and allow a *non-NP* problem to be transformed to an *NP* problem.

## **5.2 Design Automation by Learning and Evolution**

### **5.2.1 The Methodology**

Sedgewick (1988) pointed out that one way to extend the power of a digital computer is to endow it with the power of intelligent nondeterminism to assert that when an algorithm is faced with a choice of search options, it has the power intelligently to “guess” the right one. Artificially emulating Darwin’s evolutionary principle of “survival-of-the-fittest” in natural selection and genetics, the hybrid EAs developed in Chapter 2 may be applied in virtually any (non-*NP*) problems (Refer to Goldberg, 1989), since most such systems can be simulated by numerical means no matter how complex the system is, as shown in Figure 5.2 (Michalewicz, 1994). This nondeterministic algorithm requires a search time bounded by a polynomial of  $n$  (instead of an exponential of  $n$  as resulting from the exhaustive search). This methodology can thus also be used in control system design, meeting the speed and competition challenges (Li, 1996).

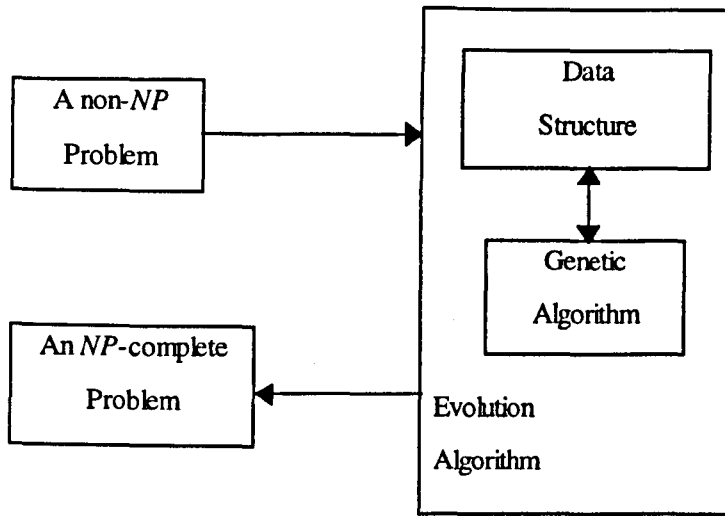


Figure 5.2 EA transforms a non-*NP* problem to an *NP* problem

Referring back to Chapter 2. As illustrated in Figure 2.1, a conventional “computer-aided control system design” (CACSD) package that provides simulation results is used to evaluate performances of candidate controllers in terms of plant outputs, closed-loop errors and control signal provision. Artificial evolution then enables CACSD to become “computer-automated control system design”, where the performances on how well the candidate controllers meet the specification are used “intelligently” to guide the coefficient adjustment.

Since the EA simultaneously evaluates  $f(P_i)$  at multiple points in the solution space  $S$ , it can easily overcome the local optimum drawback in multi-dimensional space. These points form a *population* of candidate designs as defined by:

$$P(t) = \{P_i(t), \forall i \in \{1, \dots, z\} | P_i(t) = \{p'_1, \dots, p'_n\}\} \subseteq S, \quad (5.5)$$

where  $z$  is the size of the population. The designer’s expertise or known controllers can be incorporated easily in the initial population, which will usually lead to a faster convergence and will thus overcome the *a-priori* drawback (Ng, 1995). Further, all the design criteria and practical constraints can be included by the fitness function, since this function does not need to be differentiated in a EA and simulation to evaluate for such a function can be performed.



### 5.2.2 Limitations of the Evolutionary Methodology

From the above discussions, it can be concluded that a design problem of control systems, as well as other decision making systems, can always be solved by an evolution program under the following conditions (Goldberg, 1989):

- (1) The solution space to design,  $\mathbf{S}$ , is finite or can be represented by a finite quantisation;
- (2) The system is analysable, i.e., the performance of candidate designs,  $f(\mathbf{P}_i)$ , can be evaluated; and
- (3) The performance index,  $f(\mathbf{P}_i)$ , has values with more information than a simple *True-or-False* answer. This information is needed to guided the *a-posteriori* nondeterministic evolution.

Further, for the computer-assisted design automation, *reasonable encoding schemes* (Sedgewick, 1988) must be used. An example of such a scheme is the binary coding, where the number of bits used to represent the decimal number 12 should be equal to  $\lceil \log_2(12) \rceil = 4$  and not equal to 12 itself. Otherwise the dimension of the solution space is unduly amplified exponentially.

## 5.3 Unification of LTI Control System Designs

### 5.3.1 Performance Based Design

Classical or modern linear time-invariant (LTI) control law may be in the form of the proportional plus integral plus derivative (PID), the phase lag-lead, the pole-placement, the LQR, the LQG, the  $H_\infty$  or the  $\mu$ -synthesis based control schemes. A different scheme has often to be designed using a different methodology or different algorithm. Thus, for a given application, a control engineer will face a challenge in selecting an appropriate control law from various available theories and algorithms before detailed designs are attempted. This has prompted the desire of overcoming such difficulty by unifying LTI control laws based

on performance satisfactions, as opposed to pre-selection of a specific scheme (Li *et al.*, 1995b, 1996b; Tan and Li, 1997b).

This section aims to formalise the design issue and to develop an evolutionary computation based automated design methodology. This methodology is also aimed at unifying LTI control system designs, which have the following advantages over existing individual design methods (Li *et al.*, 1995b, 1996b; Tan and Li, 1997b):

1. Unifying various individual design methods into one piece of easy-to-use design software and simplifying design engineers' tasks;
2. Automatically addressing the issues concerning stability, transient response, steady-state response, sensitivity, robustness and reliability by performance evaluations;
3. Ease of incorporating prior knowledge in the control system design by coding it in the initial population of candidate solutions, which will further reduce the design cycle;
4. Global, as opposed to local, optimisation of the control system from multiple points simultaneously in the multimodal design space;
5. Optimisation for multi-objective functions to meet various specifications with the ability of incorporating any type of constraints;
6. Automatically arriving at a number of globally optimised designs with an exponentially reduced search time compared with the exhaustive search method.

Almost all types of LTI controllers are in the form of a transfer function or a transfer function matrix equivalent to a state-space representation when the design is eventually complete. The order and the coefficients of the transfer function, however, vary with the control law or a single design objective. For example, a controller designed from the LQR scheme tends to offer a minimised quadratic error with some minimal control effort, while an  $H_\infty$  controller to offer a robust performance with a minimal mixed sensitivity function. Although the obtained coefficients or orders of these two types of controllers may be different, the common purpose of both control laws is to devise an LTI controller that could guarantee a closed-loop performance which meets certain customer specifications in either the time or the frequency domain. Therefore, a step towards unification of LTI controllers is to coin the design by meeting practical performance requirements, instead of

by a specific scheme (Li *et al.*, 1995b, 1996b; Tan and Li, 1997b), regardless of a particular domain. An uniform LTI controller as shown in Figure 5.3 can thus be described by

$$H(s) = \frac{U(s)}{E(s)} = \frac{p_n s^{n-m-1} + \dots + p_{m+2}s + p_{m+1}}{p_m s^m + p_{m-1}s^{m-1} + \dots + p_2 s + p_1} \quad (5.6)$$

where  $m < n$ . In Equation (5.6),  $\mathcal{L}^{-1}[U(s)] = u(t)$  is the controller output ‘voltage’ with usually a hard-constraint saturation range limited by the drive voltage (or current) or by a D/A converter,  $\mathcal{L}^{-1}[E(s)] = e(t)$  is the error input to the controller, the amplitude of which may also be restricted by an A/D converter, and  $p_i \in \mathbb{R}^+ \forall i \in \{0, 1, \dots, n\}$  are the coefficients to be determined in the design.

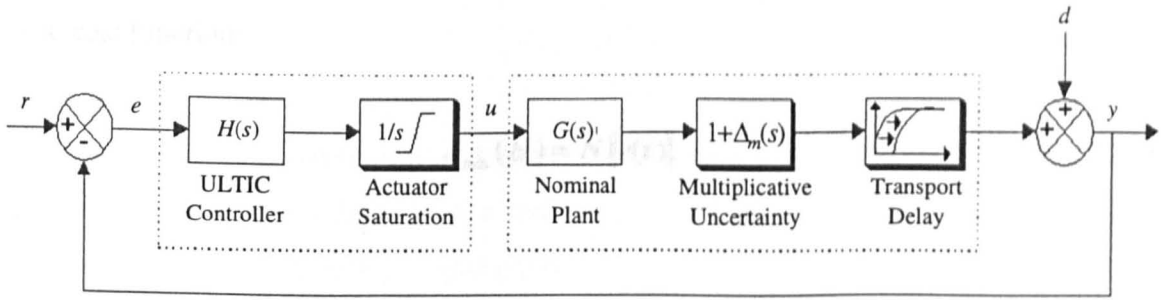


Figure 5.3 A unity negative feedback control system

For engineering applications, the controller design task is to find  $H(s)$  or  $\{p_i\}$  such that the design criteria as highlighted in Section 5.1.1 are best met. The design process involves optimisation with practical constraints such as actuator saturation, voltage limits and current limits. In a LQR design, for example, this is partially taken into account by minimising, in the time domain, a weighted term of the control energy. Clearly, the strategy of accommodating a minimal control amplitude does not precisely reflect hard constraints found in engineering practice (Li *et al.*, 1996b). Further, the multiple coefficient design space characterised by a performance index is usually multi-modal. This and practical system constraints make it inappropriate to use conventional analytical or numerical optimisation techniques to automate the design for a composite design objective or multi-objectives (Chipperfield and Fleming, 1995; Tan and Li, 1997b). Partly because of this, the

unification had not been realised until the use of evolutionary methods (Li *et al.*, 1995b, 1996b; Tan and Li, 1997b).

### 5.3.2 Building Blocks of Cost Functions for ULTIC

In this section, the feasibility of unifying classical and modern LTI control strategies in both the time and the frequency domains is reinforced, guided by performance satisfactions. The underlying aim is to let a practising engineer conveniently to obtain an “off-the-computer” controller directly from his/her building blocks of customer specifications.

#### Basic Performance Index for EA Guidance

In a design exercise, the closed-loop performance can be inverse-indexed conveniently by a basic cost function

$$J_{\min}(H) = N \|e(t)\| \quad (5.7)$$

or

$$J_{\min}(H) = \|E(j\omega)\| \quad (5.8)$$

where

$$E(j\omega) = \frac{1}{1 + H(j\omega)G(j\omega)} \quad (5.9)$$

Here  $N$  is the number of samples used for the simulation. The design task is thus to find optimal coefficients of  $H(s)$  in Equation (5.6) such that  $J_{\min}(H)$  is minimised. This is equivalent to minimising the error in either the time or the frequency domains. Here  $J_{\min}(H)$  should reflect the design criteria such as highlighted in Section 5.1.1. The following tends to reinforce this view and suggest individual building blocks if specific terms need to be emphasised.

**Observation 1:** The design of an LTI controller for an optimal performance can be unified in the time and the frequency domains. A specific LTI control scheme can be obtained from the ULTIC scheme by adding and/or multiplying the basic index of Equations (5.7) or (5.8) with a relevant specification block. Alternatively, a better and more efficient approach is to use the MOEA studied in Chapter 2 for the optimisation of these different design objectives based upon the concept of Pareto optimality.

**(I) Implicit Index to Robust Stability:** If the open-loop system is stable, then the Nyquist plot of the denominator of Equation (5.9) should not encircle its origin in any way. This means that for relatively large stability margins, the denominator plot should be relatively far away from its origin and its magnitude should have a relatively large value.

**Observation 2:** Minimising the basic index indirectly leads to robust stability and hence largely meets Spec. 1.

**(II) Improving Transients:** If suppressing overshoots and undershoots are required, weighting against the transient may be realised in either the time or the frequency domain by incorporating the following building block:

$$|e(0)| = \left| \frac{A}{1 + H(\infty)G(\infty)} \right| \quad (5.10)$$

**Observation 3:** A simple “weighting” against overshoots or undershoots is to use the  $L_\infty$  norm for the system response in the time domain. The  $L_\infty$  norm in the time domain places an emphasis on the maximum amplitude of system response occurs at the transient.

**(III) Minimising Steady-State Errors:** If suppressing errors at a particular time period needs to be emphasised in guiding the optimisation, a time or frequency weighting function similar to LQR and LQG control (Hunt, 1992), for example, may be added. Since evolution does not require direct gradient-guidance, the weighting function design becomes much more flexible. Note that if the control reference is a step of size  $A$ , then

$$|e(\infty)| = \left| \frac{A}{1 + H(0)G(0)} \right| \quad (5.11)$$

**Observation 4:** A simple weighting against the steady-state error is to incorporate the building block of Equation (5.11) in either the time or the frequency domain. Another simple “weighting” against steady-state errors is to multiply the simulation time index to the basic index in time domain, which increasingly emphasis the error while the system response towards the steady-state.

**(IV) Implicit Index to Disturbance Rejection:** Study Figure 5.3 and Equation (5.9) again. The magnitude of the transfer from the disturbance to the closed-loop output is give by

$$\left\| \frac{Y(j\omega)}{D(j\omega)} \right\| = \left\| \frac{1}{1 + H(j\omega)G(j\omega)} \right\| \|G_d(j\omega)\| \quad (5.12)$$

**Observation 5:** The disturbance rejection is maximised if the basic index is minimised. The upper limit of this disturbance rejection is however bounded by limited controller gain due to the actuator saturation.

**(V) Implicit Index to Robustness against Plant Uncertainty:** In Figure 5.3, the magnitude of the sensitivity of the closed-loop transfer function to the plant transfer function is given by

$$\begin{aligned} \|S_G^{G_c}\| &= \left\| \lim_{\Delta G \rightarrow 0} \frac{\Delta G_c(j\omega) / G_c(j\omega)}{\Delta G(j\omega) / G(j\omega)} \right\| \\ &= \left\| \frac{1}{1 + H(j\omega)G(j\omega)} \right\| \end{aligned} \quad (5.13)$$

**Observation 6:** The closed-loop sensitivity to the plant uncertainty is minimised if the basic index is minimised.

### Reconciling Accuracy and Chattering

It is known that smooth control actions often lead to steady-state errors. High control actions usually result in low steady-state errors and high robustness, but also result in chattering and excessive wear of actuators. This may be reconciled by constructing performance index blocks in a similar manner to phase lag-lead compensation or PID control, noting that the chattering is reflected by the rate of change of error. Note also that index block manipulations can be realised easily in evolutionary guidance, since it is only required by an EA to calculate  $J_{\min}$  and not its gradients. To penalise both the error and chattering at the steady-state in the time domain, weighting can be simply realised by multiplying the basic index by time. Further, weighting this way will not penalise a rapid transient.

**Observation 7:** The requirements of a high accuracy and low chattering at the steady-state can be reconciled in the time domain by adding to the basic index a building block of error derivatives and multiplying them by a building block of time as in

$$J_{\min} = \sum_{t=1}^N (e_t^2 + \dot{e}_t^2) t \quad (5.14)$$

Note that this performance index tries to eliminate chattering and oscillation of a closed-loop response, by introducing the term  $\dot{e}$ , i.e., the rate of change in error, which goes beyond those conventionally used ISE, IAE or ITAE indexes usually found in a CACSD package.

### Explicit Index for Robust Performance

To achieve robust performance of the system shown in Figure 5.3, synthesising a control law which maintains system response and error signals within pre-specified tolerances despite uncertainties will be needed. The following definitions are used to quantify the robustness of the system performance:

$$S(s) \stackrel{def}{=} \{I + G(s)H(s)\}^{-1} \quad (5.15)$$

$$T(s) \stackrel{\text{def}}{=} G(s)H(s)\{I + G(s)H(s)\}^{-1} = I - S(s) \quad (5.16)$$

The two matrices  $S(s)$  and  $T(s)$  are known as the *sensitivity function* and *complementary sensitivity function*, respectively. The singular values of  $S$  determine the disturbance attenuation since  $S$  is in fact the closed-loop transfer from disturbance  $d$  to plant output  $y$  as shown in Equations (5.12) or (5.15). A disturbance attenuation performance specification may thus be written as

$$\bar{\sigma}(S(j\omega)) \leq |W_1^{-1}(j\omega)| \quad (5.17)$$

where  $|W_1^{-1}(j\omega)|$  is the desired disturbance attenuation factor. Allowing  $W_1(j\omega)$  to depend on frequency  $\omega$  enables one to specify a different attenuation factor for each frequency  $\omega$ .

The singular value Bode plot of  $T(s)$  is used to measure the stability margins of the feedback system in face of multiplicative plant uncertainties. The multiplicative stability margin is, by definition, the “size” of the smallest stable  $\Delta_M(s)$  which destabilises the system shown in Figure 5.3.

**Small Gain Theorem** (Zames, 1966): Suppose the nominal plant in Figure 5.3 is stable with  $\Delta_M$  being zero. Then the size of the smallest stable  $\Delta_M(s)$  for which the system becomes unstable is

$$\bar{\sigma}(\Delta_M(j\omega)) = \frac{1}{\bar{\sigma}(T(j\omega))} \quad (5.18)$$

where  $\bar{\sigma}$  defines the largest singular value. Thus, the smaller  $\bar{\sigma}(T(j\omega))$  is, the greater the size of the smallest destabilising multiplicative perturbation will be and, hence, the greater the stability margins of the system. It is usual to specify the stability margin of control systems via singular value inequalities such as



$$\overline{\sigma}(T(j\omega)) \leq |W_2^{-1}(j\omega)| \quad (5.19)$$

where  $|W_2^{-1}(j\omega)|$  is the respective sizes of the largest anticipated multiplicative plant uncertainties. A control system satisfies the robust performance if both the disturbance attenuation specification in Equation (5.17) and multiplicative stability margin specification in Equation (5.19) are achieved. Figure 5.4 shows the block diagram of an ULTIC system design to achieve the robust performance and to meet Specs. 4 and 5 in Section 5.1.1. The outputs  $y_1$  and  $y_2$  are normalised by the two weighting functions  $W_1$  and  $W_2$ , which indicates that the design to meet robust performance is to optimise the  $H_\infty$  norm of the normalised outputs  $y_1$  and  $y_2$  to be less than 1. In Figure 5.4, the output  $y$  can be used to reflect the Specs. 1, 2, and 3 in Section 5.1.1.

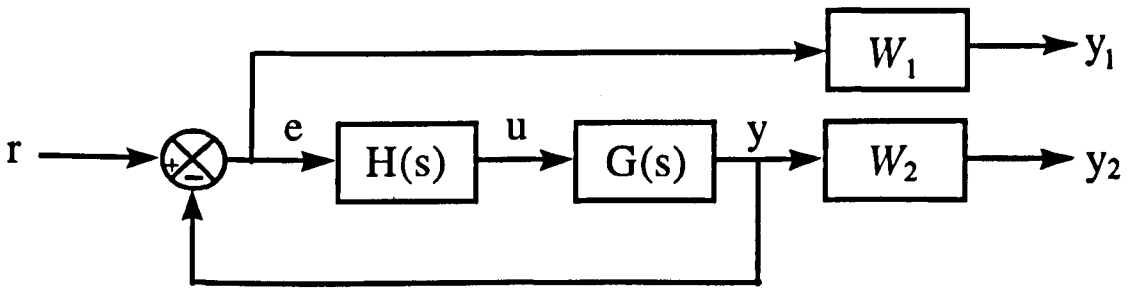


Figure 5.4 Block diagram of a robust performance design procedure in ULTIC

## 5.4 Design Automation of ULTIC Systems for Linear Plants

### 5.4.1 ULTIC for a Delayed Linear Plant

A LTI plant experimented here is a time-delayed DC servo-system for velocity control. This system was studied in Section 4.2.1.2 and is given by:

$$\ddot{\omega}(t-.06) + \left( \frac{JR + LB}{LJ} \right) \dot{\omega}(t-.06) + \left( \frac{RB}{LJ} \right) \omega(t-.06) = \left( \frac{K_T}{LJ} \right) v(t) \quad (5.20)$$

where  $v(t) \in [-5V, 5V]$  is the input field control voltage with a hard limit,  $\omega(t) \in \Re$  the angular velocity calculated from a Gray-code shaft encoder. The following model parameter values are based on manufacturer's specification.  $K_T = 13.5 \text{ NmA}^{-1}$  the torque constant for a fixed armature current,  $R = 9.2 \text{ W}$  the resistance of the field winding,  $L = 0.25 \text{ H}$  the inductance, and  $J = 0.001 \text{ kgm}^2$  the moment of inertia of the motor shaft and load. The friction coefficient of the shaft,  $B$ , changes from  $2.342 \times 10^3 \text{ Nms}$  to  $1.34 \times 10^3 \text{ Nms}$  when an eddy current brake is released.

#### 5.4.1.1 Performance Based Time-Domain Design

The hybrid EA detailed in Chapter 2 has been applied to the design of an ULTIC controller for this system. The objective is to obtain an ULTIC controller which offers a closed-loop step response with a short rise time, fast-settling time and small steady-state errors as detailed in Section 5.1.1. The performance index of Equation (5.14) is used in the EA here to achieve the requirement of good transient response and low chattering at the steady-state. Although it is unnecessary to use a third-order controller for a second-order plant, it is used to test the ability of the EA in finding an optimal and reasonable coefficient set. Here, a sampling period of 10 ms is used as the time constant for this system is relatively small. For this design, the hybrid EA has been run for 50 generations with a population size of 50, taking 45 minutes on a 50 MHz Intel 80486 processor. The search converges rapidly as shown in Figure 5.5, where both the average fitness and the highest fitness in every generation are plotted. The average fitness shows the trend of the entire population and indicates a better picture of evolution towards the optimum.

The resulting 3<sup>rd</sup> order controller at the end of 50th generation are given by

$$H(s) = 77.68 \frac{0.01s^3 + 0.31s^2 + 1.16s + 1}{s^3 + 23.82s^2 + 29.23s + 0} \quad (5.21)$$

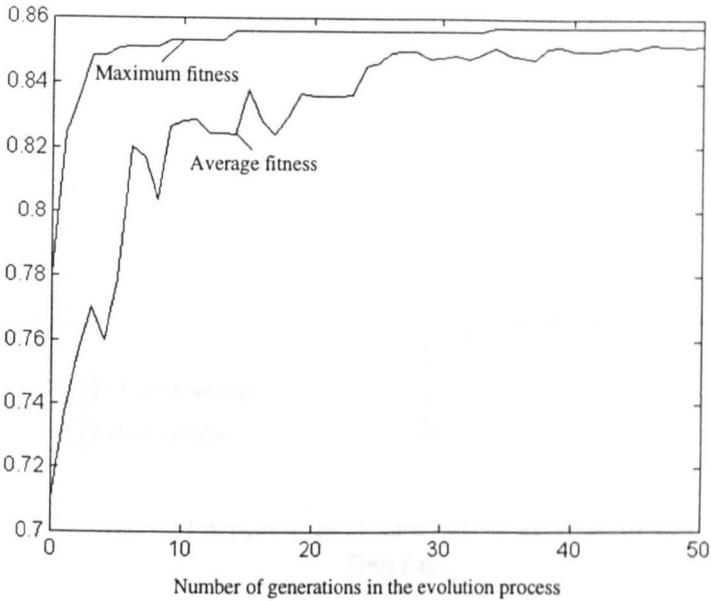


Figure 5.5 Evolution of fittest ULTIC controllers for the DC motor

It can be seen from the above values that the EA tends to provide a controller which introduces an integrator to the Type 0 system of Equation (5.20). The EA also tends to approach a 2<sup>nd</sup> order controller for this application that is to control a 2<sup>nd</sup> order plant, as the coefficients  $p_8$  (being 0.01) and  $p_4$  (being 1) are relatively small. If the order of the controller is fixed to the 2<sup>nd</sup> order in the EA search, however, the EA will recommend a controller with very high gain (Li, 1995). In order to test the EA designed controller, a reference given by:

$$r(t) = 2Au(t) - Au(t-\tau) \text{ r.p.s.} \tag{5.22}$$

is applied, where  $A = 4.5$ ,  $u(t)$  is the unit step signal, and  $\tau = 5\text{s}$ . It is also assumed that the motor suffers from a friction disturbance caused by releasing the brake at  $t = 3\text{s}$  and re-applying it at  $t = 8\text{s}$ . The captured closed-loop response of this system is shown by Curve 1 in Figure 5.6.

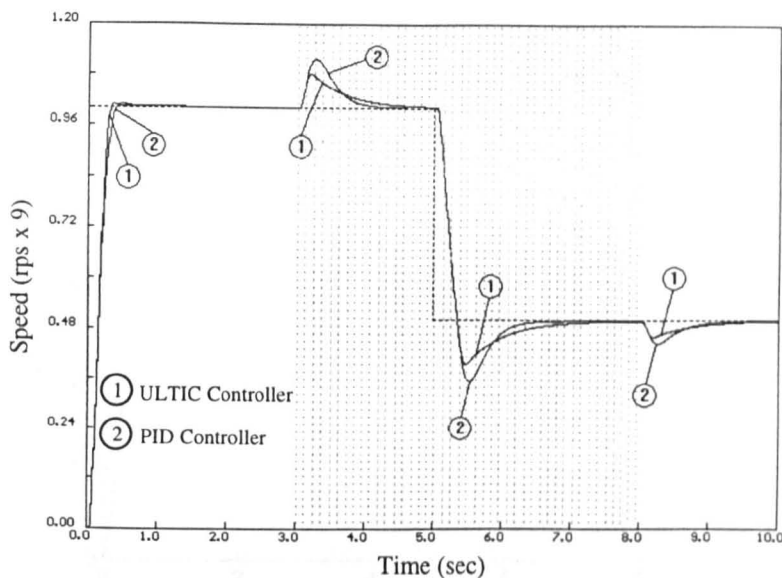


Figure 5.6 Simulated responses of the EA designed ULTIC and PID controllers, where parameter uncertainties occur at  $t = 3$  and  $8$  s

In order to compare the performance of this ULTIC controller with a PID controller, the hybrid EA is also used to search for a pure PID controller as defined by:

$$\frac{U(s)}{E(s)} = K_P + K_I \frac{1}{s} + K_D s \quad (5.23)$$

The “fittest” PID coefficients obtained under the same conditions are:

$$K_P = 0.89, \quad K_I = 2.56, \quad K_D = 0.63$$

and the response is also shown by Curve 2 in Figure 5.6. The 3<sup>rd</sup> order ULTIC controller offers a slightly better result than the PID controller as the 3<sup>rd</sup> order controller design has more degrees of freedom compared to the PID controller. To further validate the designed ULTIC controller, the closed-loop response of the true system of Equation (5.20) is shown in Figure 5.7. It can be seen that both simulated and implemented responses are consistent and robust to the friction disturbance and step down command test.

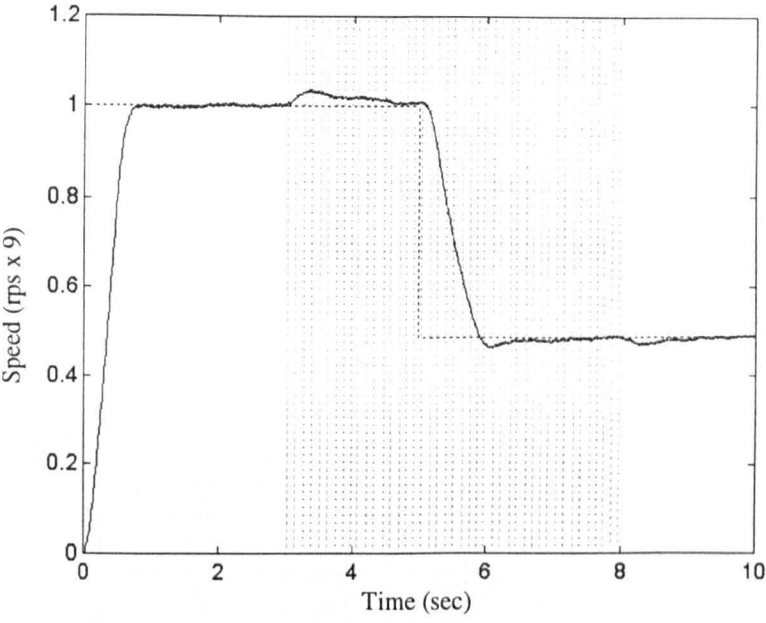


Figure 5.7 Implemented response of the EA designed ULTIC controller, where parameter uncertainties occur at  $t = 3$  and  $8$  s

5.4.1.2 Design with Emphasis on Robust Performance

Consider again the DC servo-mechanism for velocity control described in Equation (5.20). An ULTIC system is design here to achieve the robust performance using the hybrid EA detailed in Chapter 2. The objective is to obtain an ULTIC controller which satisfy both disturbance attenuation and stability margin specifications within the actuator saturation limit. The two weighting functions are chosen in the form as a low-pass filter and a high-pass filter (phase-lead compensator):

$$W_1(j\omega) = \left\| \frac{1}{1 + j\omega} \right\|_{\omega \in [0.001, 100]} \tag{5.24}$$

$$W_2(j\omega) = \left\| \frac{50(100 + j\omega)}{10000 + j\omega} \right\|_{\omega \in [1, 100000]} \tag{5.25}$$

Referring back to Section 5.3.2. Frequency response of the inverse of performance weighting function for robustness test is drawn in Figure 5.8. It can be seen that the more

the inverse sensitivity function is minimised as bounded by the weighting function  $W_1$  and the actuator saturation, the more the attenuation of the disturbance to the system. Figure 5.9 shows the frequency response of multiplicative uncertainty weighting function.

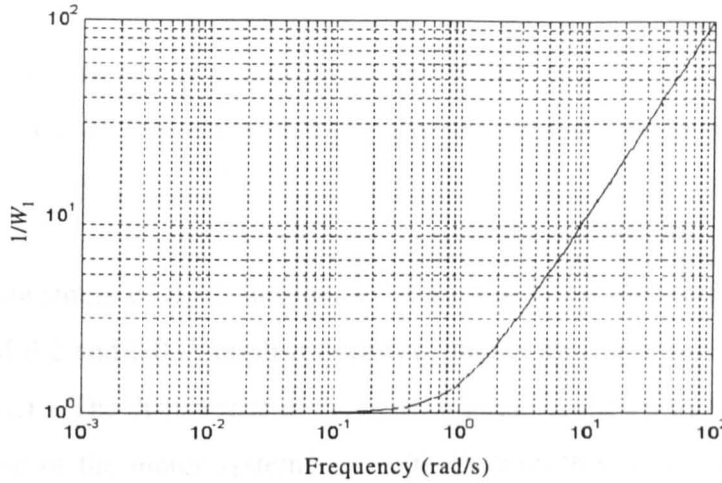


Figure 5.8 Inverse of performance weighting function

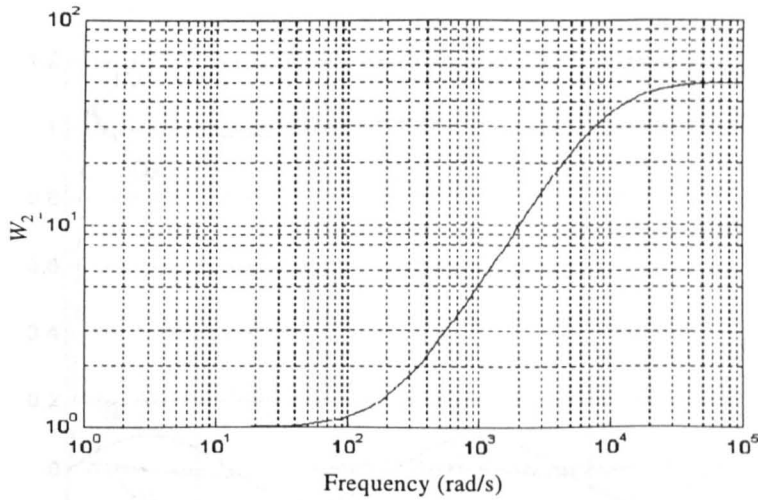


Figure 5.9 Multiplicative uncertainty weighting function

The design also needs to address the nonlinearity resulting from actuator such that the controller does not drive at saturated range, being  $[-5, 5]$  V. In order to achieve this, a third cost function is thus incorporated, which is given by

$$J_{act\_sat} = |5 - u| \quad (5.26)$$

The resulting 3<sup>rd</sup> order controller from the EA is

$$H(S) = \frac{12.46s^3 + 38.44s^2 + 44.5s + 27.68}{2.78s^3 + 3.56s^2 + 3.32s + 0} \quad (5.27)$$

The closed-loop response of this system is shown by Curve 1 in Figure 5.10. Subject to the hard voltage limit, the control action that provides this closed-loop response is shown in Figure 5.11.

To validate the robustness of the controller, a 0.2 Hz sine wave disturbance with peak-to-peak amplitude of 0.2 and 0.01 sampling period as shown by Curve 4 of Figure 5.10 was applied to the system. The disturbance at the system output is shown by Curve 2 in Figure 5.10. The response of the motor system that suffered from this disturbance is shown by Curve 3 in Figure 5.10. The responses clearly reveal the effect of the disturbance has been attenuated successfully.

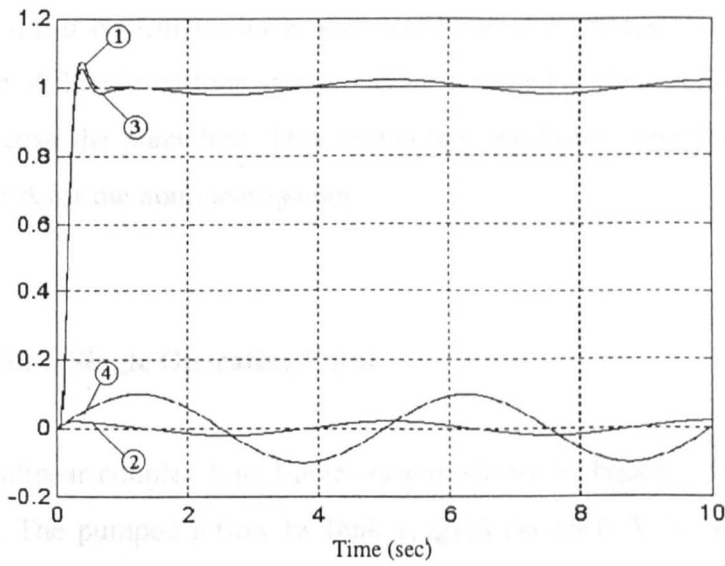


Figure 5.10 Response of the step and disturbance inputs

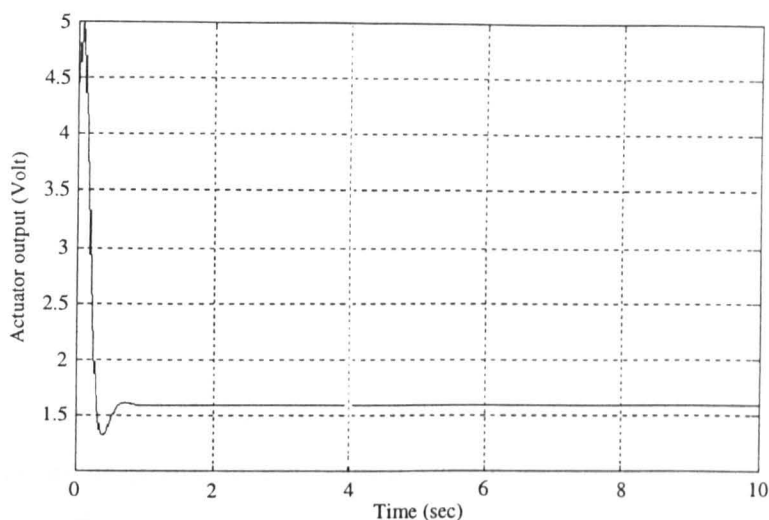


Figure 5.11 ULTIC action with an actuator constraint of 5 volt

5.4.2 ULTIC for a Nonlinear Plant Bypassing Linearisation

Since the EA optimises the controller by performance evaluations, it can be applied directly to the design of linear controllers for a nonlinear plant at a pre-specified operating point. The performance of the closed-loop system can be explored at this operating point without the need to linearise the plant first. This means that the linear controller can be directly designed by the EA for the nonlinear system.

5.4.2.1 Design for a Single Operating Point

Consider the nonlinear coupled liquid-level system shown in Figure 3.6 and described by Equation (3.17). The pumped inflow to Tank 1,  $Q_1(t)$  (in  $\text{cm}^3\text{s}^{-1}\text{V}^{-1}$ ), is the input used to control the liquid level in Tank 2,  $h_2(t)$  (in cm). In this system, there is an overall delay of 6 s from the input  $Q_1(t)$  to the output  $h_2(t)$ . The ULTIC controller needs to be designed for the operating point that sets Tank 2 level to 10 cm, with a fast response and settlement and small steady-state errors. The sampling period used is 1 s.



It took about 1 hour for the hybrid EA to evolve for 50 generations with a population size of 50, on a 50 MHz Intel 80486 processor. The evolved ULTIC controller by the hybrid EA is given by

$$H(s) = \frac{36.6s^3 + 46.3s^2 + 161.6s + 0.76}{s^3 + 0.086s^2 + 0.97s + 0} \quad (5.28)$$

The system of Equation (3.17) can be regarded as a Type 0 system. It is seen that the EA tends to supply an integrator to the control system to eliminate the steady-state error. Similarly, such a performance evaluation based method can also be applied to the search of a pure PID controller directly for the nonlinear system. For comparison, a PID controller has been evolved, whose coefficients are given below,

$$K_P = 99.5, \quad K_I = 0.81, \quad K_D = 2.07$$

The ULTIC and PID controllers are tested against the command signal given by Equation (5.22), where  $A = 0.05$  m and  $\tau = 450$  sec. The simulated performances of the closed-loop systems for the two controllers are compared in Figure 5.12. Figure 5.13 shows the implemented responses of the physical plant with the two controllers. It can be seen that both the simulated and implemented performances are promising and consistent.

To test the robustness of the designed ULTIC and PID controllers, a second inflow,

$$Q_2(t) = 8.33 [u(t-300) - u(t-600)] \text{ cm}^3\text{s}^{-1}\text{V}^{-1} \quad (5.29)$$

is used as a disturbance added to Tank 2. Again, the 3rd-order controller yields a better performance in terms of overshoot and of rejecting disturbances.

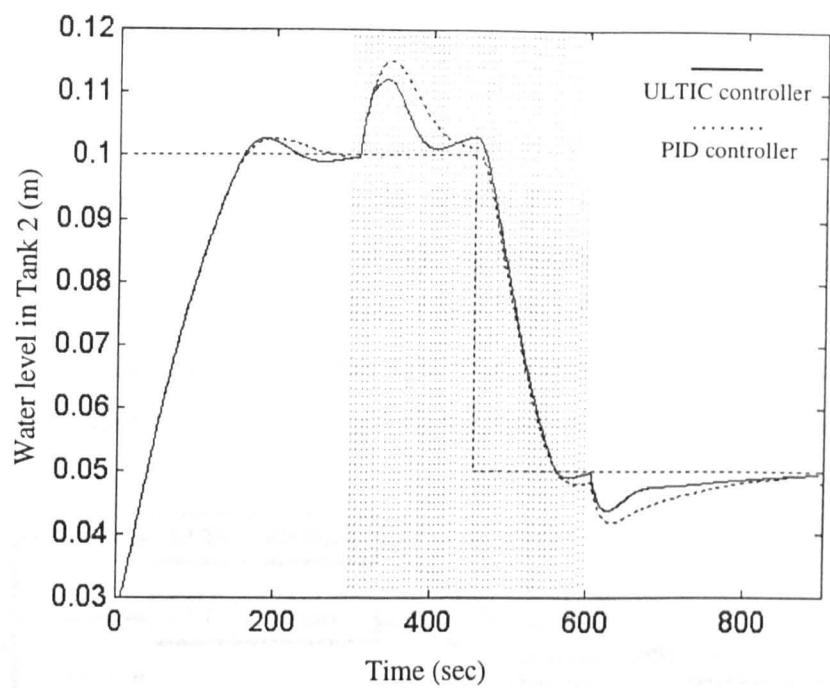


Figure 5.12 Simulated responses of the ULTIC and PID controllers for the nonlinear model

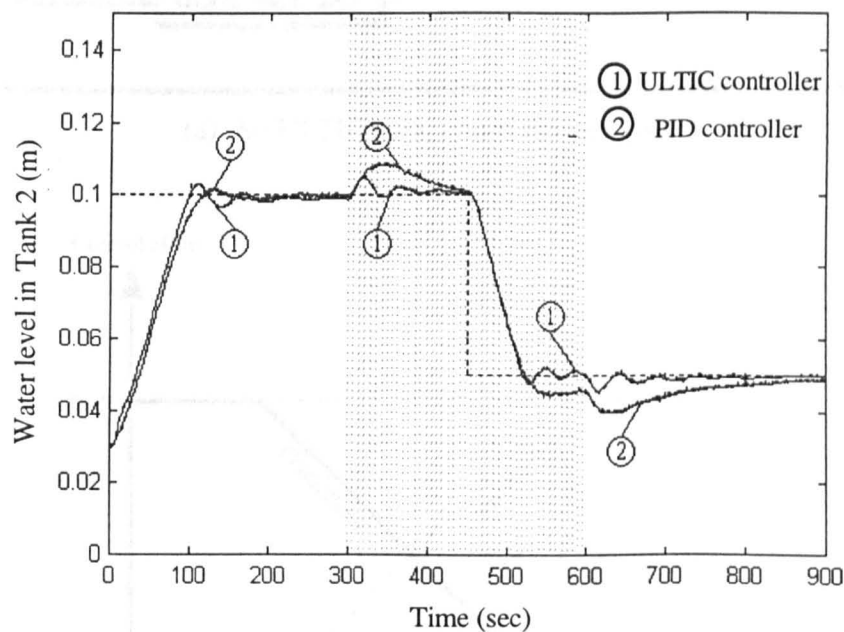
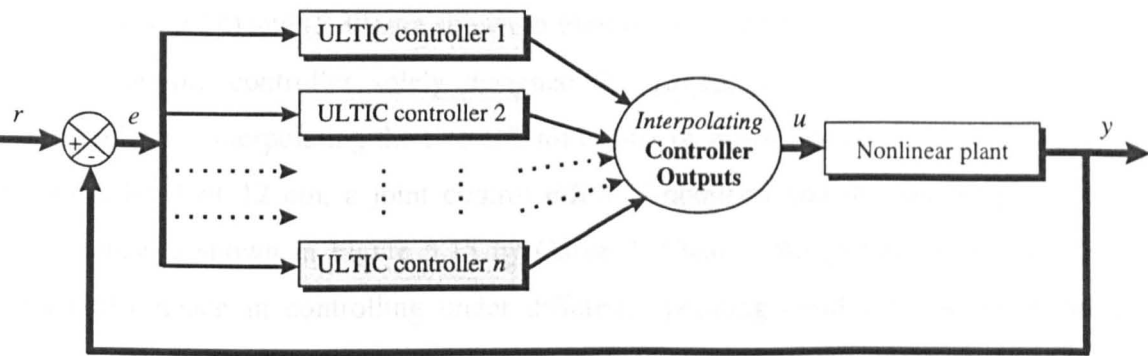


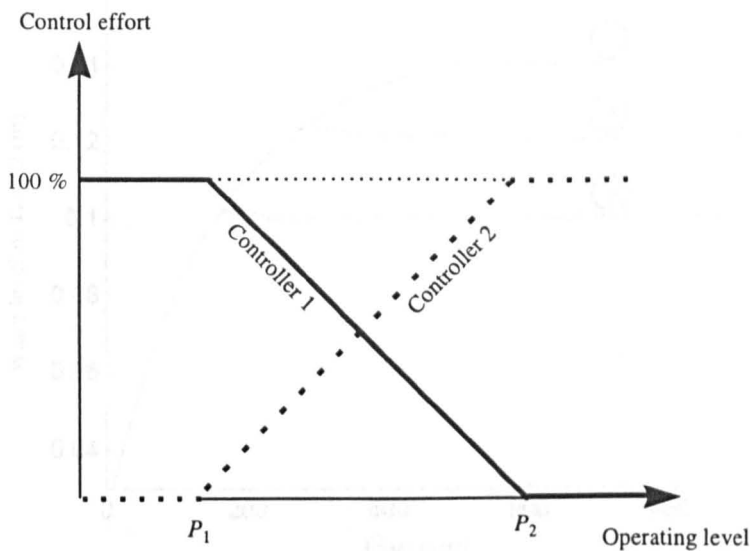
Figure 5.13 Implemented responses of the ULTIC and PID controllers for the nonlinear plant

5.4.2.2 Design for an Operating Trajectory

In many control engineering applications, controller design for a single operating point of a nonlinear plant may not be adequate. For this, the ULTIC system design will be extended to control along a trajectory of operating points. As illustrated in Figure 5.14, the approach will be similar to “gain scheduling” control, where two local linear controllers are designed based on two operating conditions of the nonlinear system. Then a global control effort is formed by joining the two local control efforts together with the ‘basis function’.



(a) An ULTIC local controller network



(b) Interpolation of local ULTIC controllers

Figure 5.14 An ULTIC local controller network system

To illustrate the methodology, another ULTIC controller has been evolved for the same nonlinear coupled liquid-level system. This is in addition to Equation (5.28) in the previous subsection and is at the operating level of 14 cm. After running the hybrid EA for 50 generations with a population size of 50, the evolved ULTIC controller is,

$$H(s) = \frac{6.4s^3 + 44.3s^2 + 125.3s + 0.24}{s^3 + 0.2s^2 + 0.77s + 0} \tag{5.30}$$

The captured closed-loop performances of ULTIC controllers with combined control effort by Equations (5.28) and (5.30) are shown in Figure 5.15. Curves 1 and 2 show the response provided by the controller solely designed for 10 cm and 14 cm operating levels, respectively. By interpolating the two control efforts of Equations (5.28) and (5.30) for an operating level of 12 cm, a joint control effort is obtained and the resulting controlled performance is shown in Figure 5.15 by Curve 3. Clearly, the global controller offers a good performance in controlling under different operating conditions of the nonlinear system.

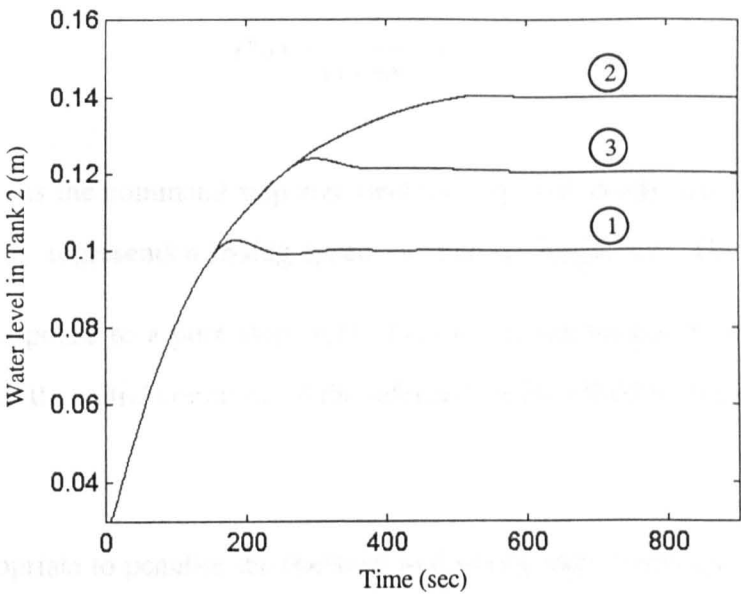


Figure 5.15 Closed-loop responses of the local and global ULTIC controllers

### 5.4.3 Model Following Control

In some control applications such as soft-start electric motor, vehicle or vessel control systems, requiring the closed-loop response to follow a discontinuous step command may be disruptive and may be impractical. In some cases, a time-varying set point is also required. In these circumstances, critically damped step response is often chosen as the reference of the system to follow, i.e., the closed-loop system is required to follow a critically-damped linear second-order system. Since ULTIC design is based on performance evaluations, it can be easily applied to these control systems. Without loss of generality, the same liquid-level control system is experimented here. As shown in Figure 5.16, the objective is to control the liquid level of Tank 2,  $h_2$  to follow a critically-damped reference:

$$\hat{y}(t) = \mathcal{L}^{-1} \left[ R(s) \cdot \frac{A}{s} \right] + H_3 \quad (5.31)$$

which is the step response of the reference model:

$$R(s) = \frac{1}{(1 + \alpha s)^2} e^{-sT_d} \quad (5.32)$$

where  $A = 0.1$  m is the command step size (and the required steady-state value). Here  $\alpha$  is set to 0.02 and  $1/\alpha$  represents a ‘rising speed’ or ‘natural frequency’. The smaller the  $\alpha$  is, the closer the response to a pure step  $\hat{y}(t)$ .  $T_d = 6$  s is the transport delay inherit in the system. Note that the initial condition of the reference is  $H_3 = 0.03$  m, the level of the outlet orifice.

Since it is appropriate to penalise the transient and steady-state errors on the same basis for a critically damped reference, the cost

$$J = \sum_{i=1}^N \left( \hat{y} - y \right)^2 \quad (5.33)$$

is adopted. The hybrid EA detailed in Chapter 2 was run with a population size of 50. The best ULTIC system evolved at the 50<sup>th</sup> generation is found to be:

$$H(s) = \frac{98.8s^3 + 996s^2 + 111s + 15.46}{s^3 + 0.43s^2 + 0.1s + 0.0} \tag{5.34}$$

The response of this closed-loop system is shown by the dotted line in Figure 5.17. The reference model output is also shown in the figure, by a solid line. It can be seen that the ULTIC system has yielded a good performance, with the output almost exactly overlaps the critically damped reference. The model following performance is significant, noting that the plant controlled is actually nonlinear, while the model followed is linear. This reinforces the power of the EA, as it reassert that such an algorithm can tune a linear controller and realise its potential in full in dealing with various applications.

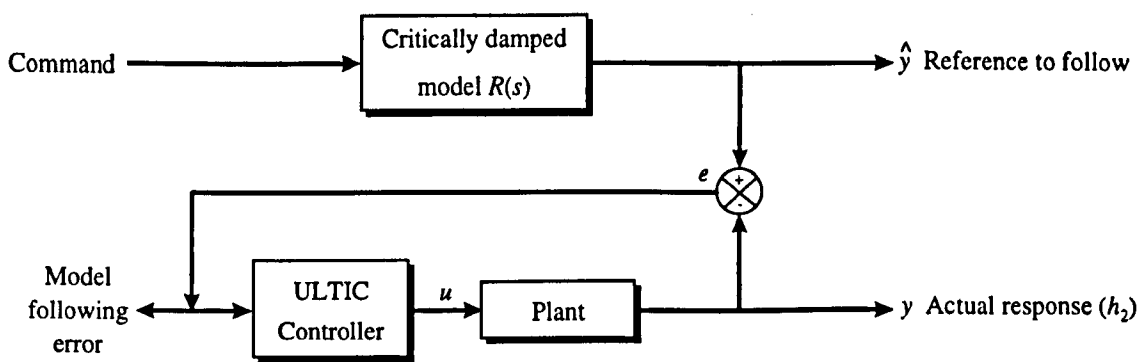


Figure 5.16 A model following ULTIC control system

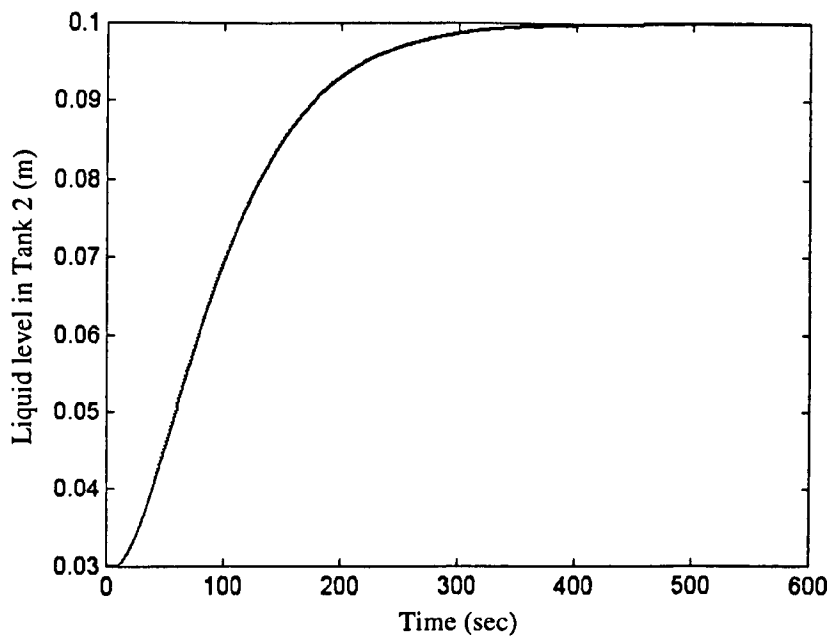


Figure 5.17 Simulated response of the EA designed ULTIC system and the critically-damped step response it follows

5.5 Multiple Input and Multiple Output ULTIC

To illustrate MIMO ULTIC system, the nonlinear model shown in Equation (3.17) is used here as a two-input and two-output system. The input to Tank 2,  $Q_2$ , is now the second system input. Liquid level in Tank 1 is desired to be 10 cm and that in Tank 2 to be 9 cm, with a minimal rise-time, overshoot and steady-state error. The MIMO control system is shown in Figure 5.18. For this, a diagonal controller would suffice (Ng, 1995), i.e., the controller has a transfer function matrix given by:

$$H = \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \end{bmatrix} \tag{5.35}$$

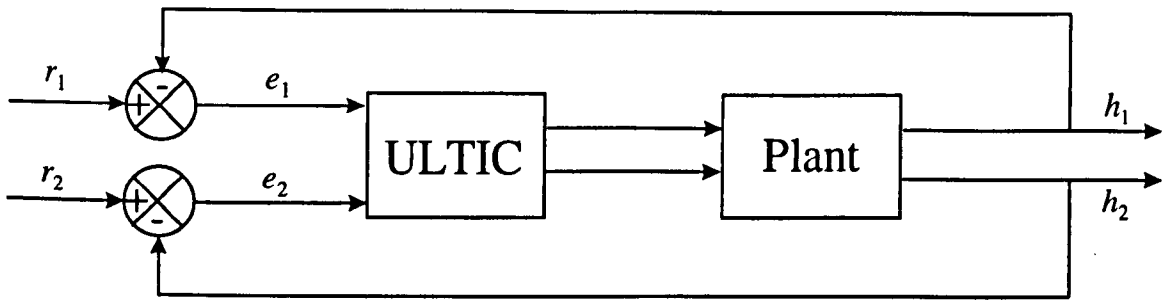


Figure 5.18 ULTIC for multiple-input and multiple-output control

Note that the steady-state value of liquid level in Tank 1 has to be specified higher than that of Tank 2 due to the requirement of outflow of liquid in Tank 1 through Tank 2 to reach the reservoir as described by Equation (3.17). Moreover, the steady-state levels of Tank 1 and Tank 2 are bounded with a maximum difference:

$$h_1(\infty) - h_2(\infty) \leq \frac{\left(\frac{Q_1}{C_1 a_1}\right)^2}{2g} \quad (5.36)$$

at the extreme of  $Q_2 = 0$  with a given  $Q_1$ . Similarly,

$$h_2(\infty) - H_3 \leq \frac{\left\{ \frac{C_1 a_1 \sqrt{2g(h_1(\infty) - h_2(\infty))}}{C_2 a_2} \right\}^2}{2g} \quad (5.37)$$

A transport delay of 1 s is found in each I/O channel of the physical system and is included in the design simulation. The hybrid EA was run with a population size of 100. The best diagonal ULTIC transfer function elements found at generation 100<sup>th</sup> are

$$H_1(s) = \frac{7.9s^3 + 30.13s^2 + 95.7s + 1.02}{1.0s^3 + 0.98s^2 + 0.73s + 0.0} \quad (5.38)$$

$$H_2(s) = \frac{4.69s^3 + 55.76s^2 + 57.56s + 0.86}{1.0s^3 + 0.46s^2 + 0.38s + 0.0} \quad (5.39)$$



This MIMO ULTIC system has been implemented with the physical system. The control signal and closed-loop responses of the system are shown in Figure 5.19 and Figure 5.20, respectively. The performance again shows a good transient and steady-state performance offered by the ULTIC evolved using the EA. The control system also copes well with the presence of the ‘untrained’ operating point at the step-down level.

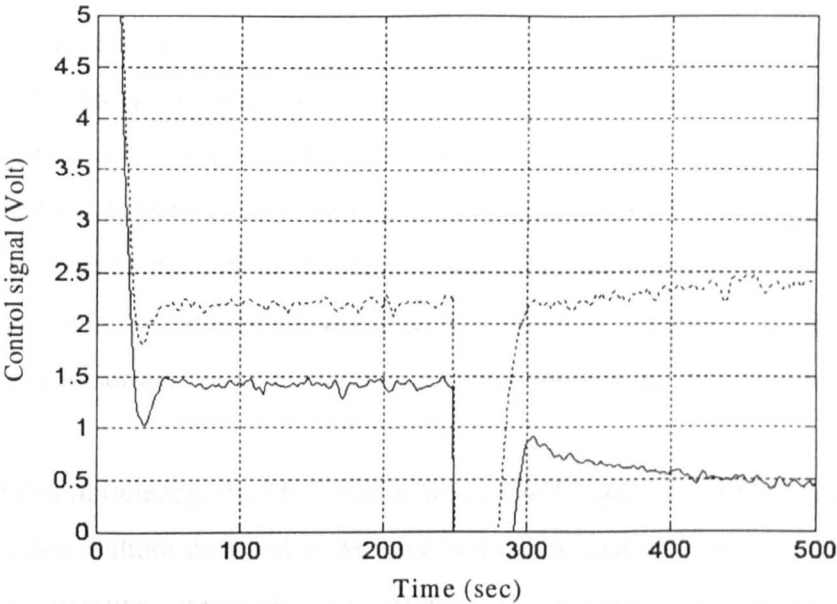


Figure 5.19 Control signal of Tank 1 { — } and Tank 2 { - - - } for the MIMO ULTIC system

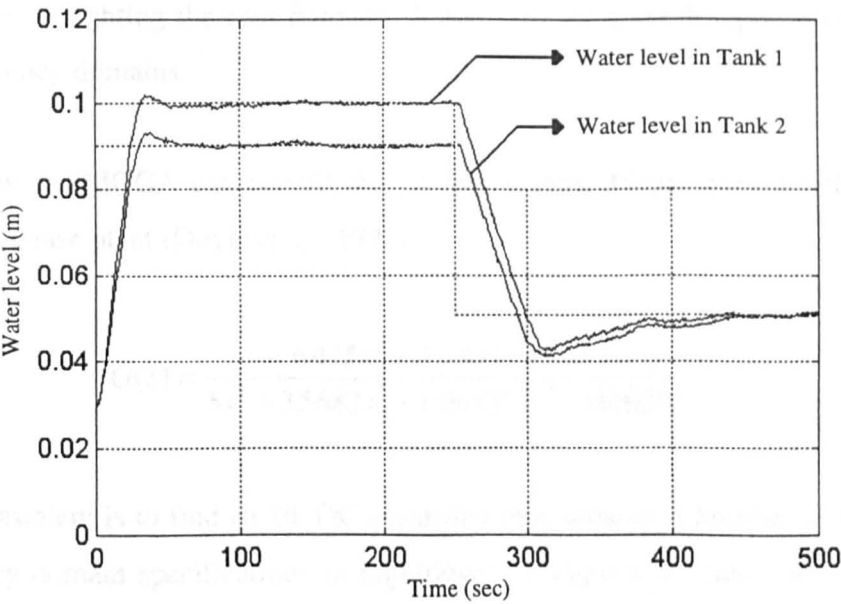


Figure 5.20 Performance of the implemented MIMO ULTIC system

## 5.6 Multi-Objective ULTIC Systems

So far, the ULTIC system design examples described in this Chapter have been based upon a composite cost function or design objective. For this, single objective evolutionary algorithms are used such that the objective function that weights individual design criteria is minimised. Although such an approach has always resulted in a final controller that best satisfies all the design specifications in the sense that the combined weighted objective function is minimised, it lacks the flexibility to interplay with the different design objectives. Using such a single combined objective function, the designer would not be able to learn which objective has been under or over penalised while the EA evolves and would have to commit himself to the design objectives beforehand. Also, for the EA to evolve a good controller, the combined objective function requires a precise settings of weights, which can sometimes be subjective and may not be easy to manage or understand.

To overcome this deficiency, the MOEA that studied in Chapter 2 has been incorporated in the ULTIC design without the need to define a composite scalar objective function. Unlike the weighted objective approach, the MOEA incorporates the concept of Pareto's domination to evolve a family of non-dominated controllers along the Pareto optimal frontier. Further, each of the individual components in the cost function can have different priorities or preferences to guide the optimisation from individual design specifications rather than pre-weighting the cost function. It can also integrate designs in both the time and the frequency domains.

To show how the MOGA can benefit the ULTIC system design, consider the following non-minimal phase plant (Doyle *et al.*, 1992):

$$G(s) = \frac{-6.475s^2 + 4.0302s + 175.77}{5s^4 + 3.5682s^3 + 139.5021s^2 + 0.0929s} \quad (5.40)$$

The design problem is to find an ULTIC controller that satisfies a number of time domain and frequency domain specifications as highlighted in Figures 5.3 and 5.4. The gains of  $W_1^{-1}$  and  $W_2$  for  $S(s)$  and  $T(s)$  are shown in Figure 5.21. Table 5.1 details the nine design objectives for a step input command. The underlying aim of setting the priority vector in

the fourth column of Table 5.1 is for obtaining a controller that first stabilises the system within the actuator saturation limit. Then, the system must also be robust to unstructured plant uncertainty and disturbance attenuation under certain level of tolerances defined by the weighting functions of  $W_1$  and  $W_2$ , respectively. While fulfilling these requirements, the system must also satisfy time domain specifications defined by the transient and steady-state responses. Some of these specifications can be conveniently made by a practising engineer using graphical means, such as the boundaries of the clear area shown in Figure 5.22, which the control engineer may be more familiar than mathematical indices.

Although determination of the objective and the priority vector may be a subjective matter and depends on the performances requirement, ranking the priorities may be unnecessary and can be ignored for a ‘minimum-commitment’ design (Guan and MacCallum, 1996). If, however, an engineer opts to commit himself to prioritising the objectives, it is a much easier task than weighting the objectives. It is obvious that other design specifications such as gain margins, phase margins and noise rejection (can be quantified by distinctive LQG or  $H_2$  norms) may also be added to the design if necessary.

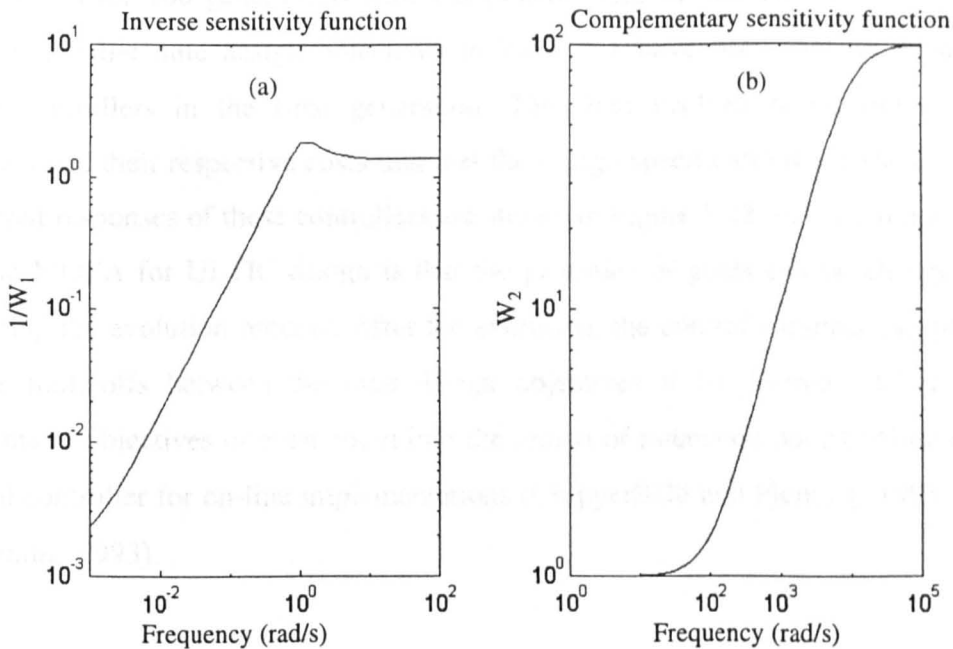


Figure 5.21 Frequency responses of (a)  $W_1^{-1}$  and (b)  $W_2$

Table 5.1. Time domain and frequency domain design specifications

<i>Customer specifications</i>		<i>Objective</i>	<i>Priority</i>	<i>Goal</i>
<b>Frequency domain</b>	1. Stability (Closed-loop poles)	$Nr([eig(A_{clp})] > 0)$	4	0
	2. Closed-loop sensitivity	$\bar{\sigma}[S(j\omega)]$	2	1
	3. Disturbance rejection			
	4. Plant uncertainty	$\bar{\sigma}[T(j\omega)]$	2	1
<b>Time domain</b>	5. Actuator saturation	$Max(u)$	3	0.5
	6. Rise time	$T_{rise}$	1	4
	7. Overshoots	$O_{shoot}$	1	0.05
	8. Settling time	$T_{settling}$	1	7
	9. Steady-state error	$SS_{error}$	1	0.01

In the evolution, the order of candidate controllers are not fixed and can also be included as another distinctive objective. Here, such an objective is felt unnecessary, as long as the maximum order is limited, which is 3 in this example. It took less than 1 hour for the MOEA to run for 100 generations with a population size of 100 on a Pentium 100MHz processor. All the nine design objectives in Table 5.1 have been met by a number of evolved controllers in the final generation. The final evolved non-dominant ULTIC controllers and their respective costs that met the design specifications are shown in Table 5.2. Output responses of these controllers are shown in Figure 5.22. An additional merit of using the MOEA for ULTIC design is that the priorities or goals can be changed at any time during the evolution process. After the evolution, the control engineer can preferably examine trade-offs between the nine design objectives if so desired, including other constraints or objectives or even zoom into the region of interested points before selecting one final controller for on-line implementations (Chipperfield and Fleming, 1995; Fonseca and Fleming, 1993).

Table 5.2 The non-dominant ULTIC controllers evolved from the MOEA

Controller	$H(s)$	Obj	Obj	Obj	Obj	Obj	Obj	Obj	Obj
		1	2, 3	4	5	6	7	8	9
1	$\frac{1.17s^3 + 7.92s^2 + 9.88s + 0.002}{5.2s^3 + 18.4s^2 + 32.28s + 20.62}$	0	0.9924	0.9993	0.3763	3.4	0.0277	7	0.0027
2	$\frac{1.24s^3 + 7.86s^2 + 9.88s + 0.001}{6.8s^3 + 18.92s^2 + 32.28s + 20.67}$	0	0.9945	0.9991	0.3617	3.2	0.0415	7	0.0091
3	$\frac{3.56s^3 + 7.9s^2 + 9.89s + 0.001}{9.36s^3 + 22.6s^2 + 32.28s + 20.6}$	0	0.9902	0.9991	0.3807	3.3	0.0439	7	0.0094
4	$\frac{1.25s^3 + 7.9s^2 + 9.9s + 0.002}{5.36s^3 + 18.4s^2 + 32.3s + 20.7}$	0	0.9928	0.9992	0.3746	3.4	0.0260	7	0.0035
5	$\frac{2.26s^3 + 8s^2 + 9.9s + 0.001}{6.8s^3 + 19.9s^2 + 32.08s + 20.62}$	0	0.9911	0.9991	0.3653	3.4	0.0281	7	0.0053
6	$\frac{1.25s^3 + 7.9s^2 + 9.8s + 0.002}{5.2s^3 + 18.4s^2 + 32.3s + 20.6}$	0	0.9948	0.9994	0.3765	3.5	0.0246	7	0.0031
7	$\frac{1.22s^3 + 7.92s^2 + 9.9s + 0.002}{5.4s^3 + 18.42s^2 + 32.28s + 20.67}$	0	0.9948	0.9993	0.3763	3.3	0.0253	7	0.0029
8	$\frac{1.25s^3 + 7.9s^2 + 9.9s + 0.001}{5.1s^3 + 18.42s^2 + 32.08s + 20.6}$	0	0.9901	0.9991	0.3775	3.4	0.0280	7	0.0021
9	$\frac{3.5s^3 + 7.91s^2 + 9.8s + 0.001}{9.3s^3 + 23s^2 + 32.3s + 20.62}$	0	0.9902	0.9991	0.3807	3.3	0.0484	7	0.0089

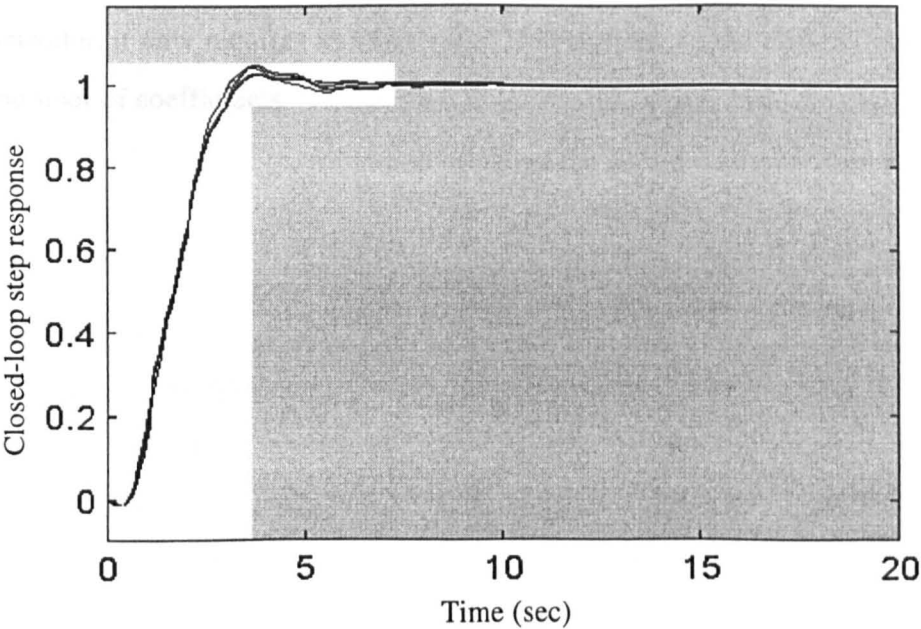


Figure 5.22 The MOEA optimised output responses

## **5.7 Parallel EA and Near-Linear Pipelinability**

As discussed in Chapter 2, an EA is naturally suited for parallelism due to its feature of multiple search points and multiple candidate solutions. Here, the ULTIC system design tasks are equally shared by up to 15 T8 transputers in a 2-D array in a Parsytec SuperCluster. Another transputer is used as the host for communications and supervisory tasks. Parallel C is used under the PARIX (PARallel unIX) operating system that offers straightforward software-channels for inter-transputer communications. To assess the effectiveness of the parallelism in ULTIC system design, the EA design process has been repeated several times on 1, 3, 9 and 15 slave transputer(s), respectively. The average speedup is plotted in Figure 5.23. It can be seen that a near-linear pipelinability is evident, which implies that evolutionary algorithms are indeed naturally suitable for parallel processing.

Another advantage of EAs is the non-deterministic polynomial (NP) feature, which implies that designing a more sophisticated controller would not take exponentially but polynomial more time than designing a simpler one. To confirm this, the design of a three-coefficient pure PID digital controller has been repeated on the same numbers of transputers. The speedups are also shown in Figure 5.23. It can be inferred that, although the number of coefficients of the controller is more than doubled from 3 of the PID to 7 of the third-order ULTIC controller, it only requires an  $O(n) \approx n \times 25\%$  increase in the design time, where  $n$  being the number of coefficients.

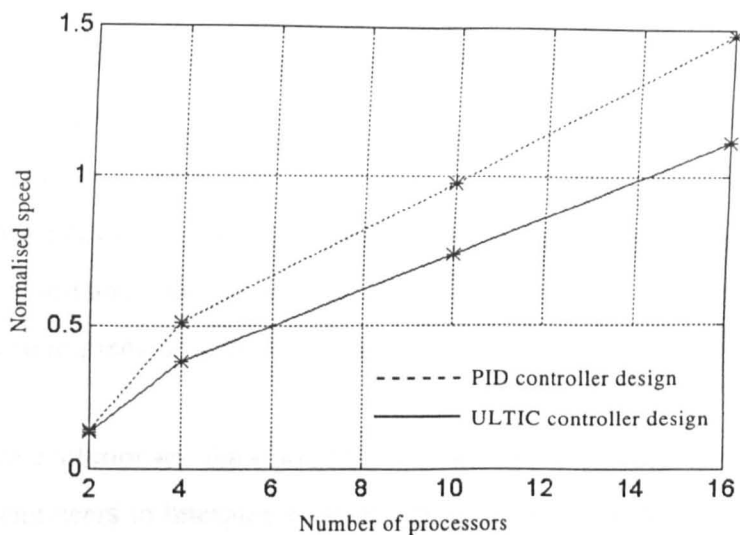


Figure 5.23 The near-linear pipelinability and NP feature of the parallel EA

## 5.8 Summary

This Chapter has developed a uniform definition of linear control system design problems using a vector space and its mapped uniform design criterion under performance/specification satisfaction. Design difficulties by conventional analytical and numerical means have been discussed. Associated problem-classification was presented, together with a possible automation technique enabled by evolution. It transforms conventionally non-polynomial (exponential) control system design problems to solvable NP-complete search problems using the EA which enables design automation in polynomial time. Drawbacks in conventional design techniques are overcome and design challenges are met by trading off precision slightly for improved tractability, robustness and ease of design.

The unification of LTI control approaches have been shown to be feasible in both the time and the frequency domains under performance satisfaction. Such a performance-based ULTIC technique can be easily realised without manual calculations or *a-priori* knowledge. However, experience gained in manual designs can be included in the formation of the initial population of candidates for improvement, which usually leads to a faster convergence. Further, this EA-based method can accommodate many requirements imposed by practical specifications, since it does not require differentiation of the

performance index. It is also applicable to the direct design of a linear controller for MIMO and nonlinear plants without the requirement of linearisation. Further, this technique has been extended to control along a trajectory of operating points of a nonlinear system and is not restricted to one local point. Apart from following a fixed step reference or set-point, time-varying references and critically damped model can also be followed by an ULTIC system using performance satisfaction based design. In addition, the resulting ULTIC systems are easy to implement with only minor storage and computational overheads.

A multi-objective evolutionary algorithm has been incorporated in the ULTIC design. This enables control engineers to interplay with design objectives before the final commitment. This evolutionary MOEA approach allows the ULTIC design to be visualisable and to satisfy the 'minimum-commitment' principle at the CAD stage.

A parallel EA has been implemented to assess the effectiveness of the parallelism in ULTIC design. It has shown that the evolutionary algorithms are indeed naturally suitable for parallel processing with a speedup of near-linear pipelinability processing time. It has also been shown that the NP convergence feature of the EA does help with an ULTIC system design which is more sophisticated than a PID controller design.



## Chapter 6

## TOWARDS AUTONOMY: EVOLVING CONTROLLERS DIRECTLY FROM OPEN-LOOP RESPONSE DATA

### 6.1 Step Response Data Represent a High-Fidelity Model

In many applications, step response data are often obtained when testing or setting the operating point. An LTI model of the plant is then identified or refined from the I/O data before the design of a controller is attempted. An example of plant response data,  $y_s(t)$ , to a step input of amplitude  $A = 2.5$  V are plotted in Figure 6.1. It is interesting to note that the step response data were, in fact, obtained from the Tank 2 output  $y(t)$  of the nonlinear plant shown in Figure 3.6 and is described by Equation (3.17). The input flow of the system was  $Q_1 = 7.0 \times 10^{-6} \text{ m}^3 \text{ s}^{-1} \text{ V}^{-1}$  and  $Q_2$  was set to zero.

Owing to the simplicity and an acceptable accuracy, a first-order plus delay identification technique is often employed to fit data from a plant. The plant may even be internally nonlinear. Partly, this is because many nonlinear plants exhibit the “Type 0” behaviour of an equivalent linear system, where a non-zero control energy is needed to maintain the steady-state operating point as indicated by Figure 6.1. Using the data plotted in Figure 6.1 and the method described by Åström and Wittenmark (1989), a first-order model with transport delay is identified as

$$G(s) = \frac{K}{1 + s\tau} e^{-T_d s} \quad (6.1)$$

with  $K = 0.01$ ,  $T_d = 0.93$  s and  $\tau = 85.1$  s. To validate the first-order model of the nonlinear system described by Equation (3.17), its response to the same 2.5 V step has been obtained and also plotted in Figure 6.1. It can be seen that the discrepancy between the model and the plant is small, but cannot be eliminated due to the limited order of the model.

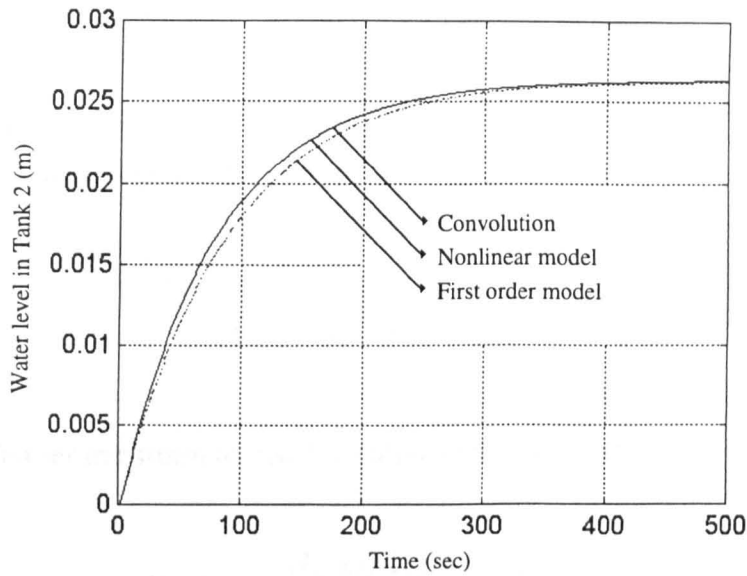


Figure 6.1 Response to a 2.5 V step input of a nonlinear plant model, response of the first-order model and response reconstructed from convoluting the plant impulse response

This discrepancy may, however, be eliminated if an infinite-order ‘linear’ model is used. To proceed, the plant unit-impulse response data can be obtained from the step response as given by:

$$g(t) = \dot{y}_s(t)/A \tag{6.2}$$

This may be regarded as such an infinite-order “model”, since convoluting this with the step input will yield a high fidelity reconstruction of the step response, as indeed shown in Figure 6.1. Note that, however, the “model” may only be valid for a consistent operating point, because the steady-state gain of the nonlinear plant is not as a constant as found in a linear plant. Such observation opens a way of designing LTI controllers directly from plant step response data (Cluett and Wang, 1991; Li *et al.*, 1996b). Of course, a more stimulating input whose spectra covers the plant bandwidth should reflect the dynamics of a practical plant more accurately. Note that this “modelling” approach may also apply to nonlinear plants for a given operating point, although a more accurate I/O relationship could be obtained by using the steady-state equilibrium and perturbing the plant round this point as adopted in the linearisation techniques developed earlier.

## 6.2 Evaluating ULTIC System Performance from Plant Step Response Data

For an unity negative feedback control system, the closed-loop output,  $y(t)$ , resulting from the control signal,  $u(t)$ , is given by

$$\begin{aligned} y(t) &= u(t) * g(t) = u(t) * \dot{y}_s(t) / A \\ &= [r(t) - y(t)] * h(t) * \dot{y}_s(t) / A \end{aligned} \quad (6.3)$$

In Laplace or Fourier transform terms, this output can be evaluated by

$$Y(j\omega) = \frac{H(j\omega)G(j\omega)}{1 + H(j\omega)G(j\omega)} \cdot R(j\omega) \quad (6.4)$$

where

$$G(j\omega) = \frac{j\omega Y_s(j\omega)}{A} \quad (6.5)$$

Thus, given an open-loop step response, the spectra of the step response or the frequency response of a plant, the performance of an LTI controller can always be evaluated in either the time by Equation (6.3) or the frequency domain by Equation (6.4) without the need of a model of the plant.

## 6.3 Evolving an ULTIC System for a Linear Plant Directly from Open-Loop Response Data

In this Section, step response data and Equation (6.3) are used with the hybrid EA detailed in Chapter 2 to evolve candidate ULTIC controllers towards performance satisfactions. The DC servo-mechanism for velocity control described by Equation (4.4) is experimented here. Step response data sampled from physical system of the DC servo-mechanism is shown by the solid line in Figure 4.5. The performance index of Equation (5.14) is used in the EA here to achieve a closed-loop step response with short rise time, fast-settling time and small steady-state errors. A third-order ULTIC controller has been evolved directly

from these step response data. After running the hybrid EA for 50 generations with a population size of 50, the resulting controller transfer function is given by

$$H(s) = \frac{7.2s^3 + 153.2s^2 + 426.9s + 293.8}{1.0s^3 + 27.6s^2 + 29.2s + 0.0} \tag{6.6}$$

The coefficients in the numerator appear to be different from those obtained using the system model as given in Equation (5.21). It is however expected, as the EA is a non-deterministic algorithm and only recommends a controller that, with whatever coefficients, offers globally on near-globally optimised performance. This observation is validated by Figure 6.2, which shows the step response of the system controlled by Equation (6.6). Note that the step-down was tested while the plant parameter value,  $B$ , was varied within the shaded area. This is to test the controller’s capability to reject disturbances or uncertainties, which had not been modelled for use with the EA based design. The response confirms that the ULTIC approach does yield a good transient and steady-state performance, with some robustness against the plant uncertainties.

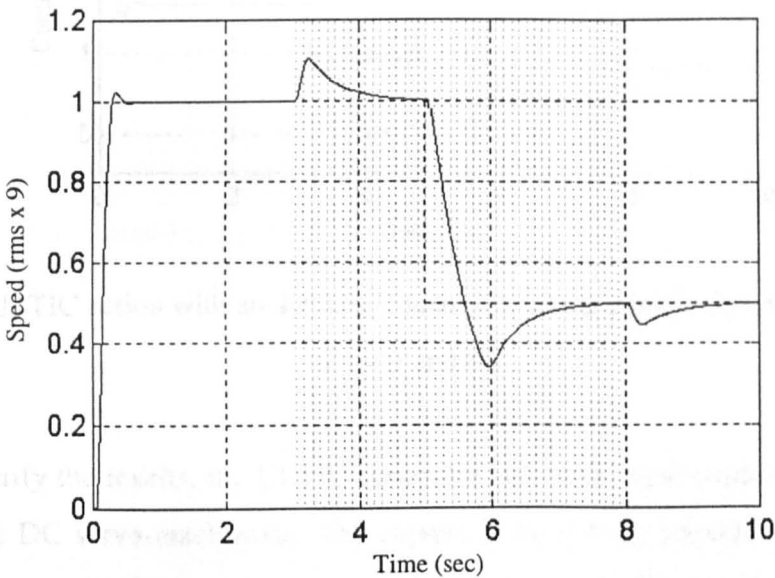


Figure 6.2 Performance of the ULTIC evolved from I/O data of an LTI plant, where parameter uncertainties occur at  $t = 3$  and  $8$  s

Again, it is not surprising to note that the performance base ULTIC system design offers an integrator automatically to this Type 0 plant. Recall that the same recommendation of using an integrator was made by the EA when a model is used for a performance based design in Equation (5.21). Both the model and I/O data based designs have recommend a relatively small coefficient for the third-order term in the numerator when a third-order controller was asked for. Subject to hard voltage limit, the control action that provides the above closed-loop response is shown in Figure 6.3. It can be seen that the feasibility of incorporating such a practical constraint in the evolutionary design not only yields a practical control signal that offers the optimised performance, but also eliminates the need of artificially optimising the control energy.

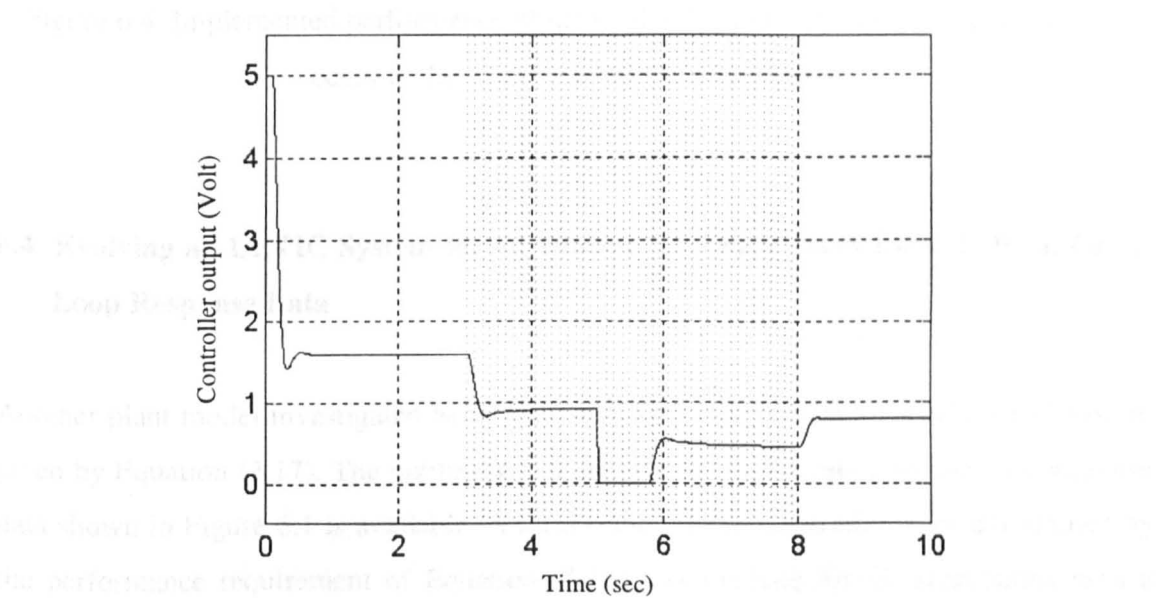


Figure 6.3 ULTIC action with an actuator constraint, where parameter uncertainties occur at  $t = 3$  and  $8$  s

To further verify the results, the ULTIC controller has been implemented for the physical system of the DC servo-mechanism. The captured closed-loop response of the system is shown in Figure 6.4. The response confirms that the plant I/O based ULTIC design is consistent for both the simulated and implemented performance, with a good transient and steady-state performance and some robustness against the plant uncertainties.

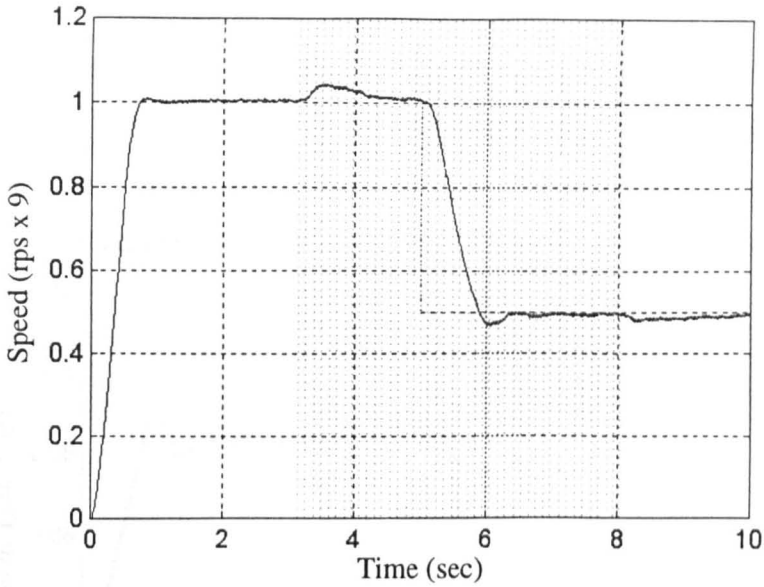


Figure 6.4 Implemented performance of the evolved ULTIC, where plant uncertainties occur at the boundaries of the shaded area

### 6.4 Evolving an ULTIC System for an Unseen Nonlinear Plant Directly from Open-Loop Response Data

Another plant model investigated here is the nonlinear coupled-liquid level control system given by Equation (3.17). The nonlinearity is unseen by the EA and only the step response data shown in Figure 6.1 is available. A third-order ULTIC controller to be determined by the performance requirement of Equation (5.14) was evolved for 50 generations with a population size of 50. The evolved ULTIC controller by the EA is given by

$$H(s)=\frac{243s^3+151s^2+273s+1.73}{1.0s^3+1.82s^2+0.44s+0.0}\tag{6.7}$$

To compare with the I/O based approach, another third-order controller was designed from the identified first-order model given by Equation (6.1). The resulting transfer function is

$$H(s)=\frac{217s^3+190s^2+299s+1.44}{1.0s^3+1.81s^2+0.4s+0.0}\tag{6.8}$$

The performances of controlling the physical nonlinear system by these two controllers have been tested at two operating points with additive disturbances. The closed-loop step responses are shown in Figure 6.5. It can be seen that the LTI controller designed without a model offers a slightly better performance in controlling the nonlinear system than that offered by the controller designed from the identified model given by Equation (6.1).

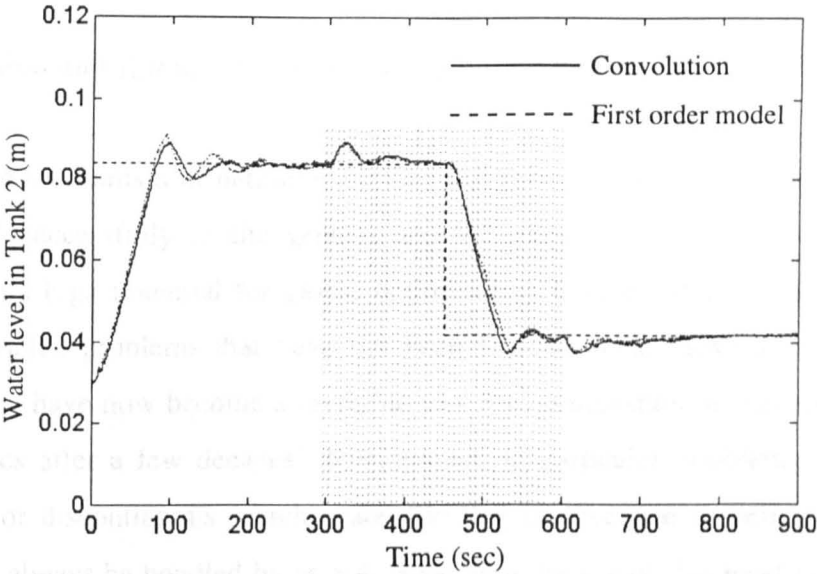


Figure 6.5 The implemented performances of the ULTIC controllers designed from the I/O data and the first-order model, where disturbance occur at  $t = 300$  and  $600$  sec

### 6.5 Summary

A direct method for control system design from plant step response data has been developed. Results obtained have shown that the design can be automated by efficient evolution from plant step response data, bypassing the system identification stage. Using the proposed ULTIC approach, control engineers only need to feed the CACSD system with the plant I/O data and customer specifications optimally to obtain an “off-the-computer” controller. This offers a step towards autonomy in building control systems.

The ULTIC control and design strategy has been validated against linear and nonlinear plants. It has been observed that the performance of the controller evolved from the response data of an internally nonlinear plant is slightly better than that designed from an identified model.

## Chapter 7

# CONCLUSION AND FURTHER WORK

## 7.1 Application and Enhancement of Evolutionary Algorithms

Emulating the mechanism of natural evolution and genetics, evolutionary algorithms have been applied successfully to the general area of control system engineering. Inherent parallelism and high potential for global optimisation, EAs are able to deal with a wide range of complex problems that have not been feasible with classical or deterministic methods. They have now become a favourite tool for optimisation in engineering, science and economics after a few decades' development. In particular, problems that exhibit a multi-modal or discontinuous search space and that involve non-numeric or mixed-type variables can always be handled by an EA. Such algorithms can also meet multi-objective criteria simultaneously and provide feasible automated solutions in control system modelling and design. Their capability in solving complex real-life problems should forecast a fruitful flourishing in the near future.

In this thesis, local interactive fine-learning realised by Boltzmann selection is incorporated in EAs to enhance their performance. Benchmark testing in this thesis has shown that the enhanced EA is accurate, effective, efficient and reliable. Pareto's optimality cost-assignment with tournament selection has also been developed with niching and mating restriction. It has been shown to be particularly useful and effective in evolving a family of non-dominated solutions along the Pareto front for multi-objective optimisation. By incorporating a priority vector, the MOEA allows different preference of the objective components and makes the optimisation more visualisable and transparent.

The effectiveness of an EA is typically characterised by its *control parameters*, such as mutation rate, crossover rate and population size. Other features such as the size of generation gap, types of crossover technique, reproduction strategy, declaration of fitness landscape, distance of niche count, selection of Boltzmann learning rate and the number of



migration individuals also have effects on the rate of convergence in the EA. Although the setting of these control parameters may depend on user's experience and prior knowledge about the problem on hand, the parameters could be reduced if "Elitist Direct Inheritance" technique developed in this thesis is employed. "One-integer-one-parameter" coding proposed in this thesis has substantially reduced the memory usage and quantisation error while having faster processing time. This coding scheme is particular useful to those optimisation applications for which accuracy is also a major concern.

## **7.2 EA Based Methodologies for Model Reduction and Linearisation**

This thesis has developed a generic model reduction technique in both the time and the frequency domains using the EA proposed in Chapter 2. Reduced models for both discrete and continuous-time systems could be obtained by minimising  $L_2$  or  $L_\infty$  norm as desired. The technique is applicable to both SISO and MIMO systems and has provided a tighter  $L_\infty$  norm error bound than existing methods. Model reduction examples studied in this thesis also show that the evolutionary optimising method provides a better performance to that of existing methods.

Evolution based techniques for multivariable  $L_2$  and  $L_\infty$  norm linearisation has also been studied. The linearisation is performed based on the plant input-output behaviour in the time or the frequency domain. The techniques have also been extended to allow linearisation for an entire operating region by linear approximate-model networks.

Enabled by a control gene as a structural switch, the evolutionary method is applied to recommend both an optimal order number and optimal corresponding parameters simultaneously. Although only  $L_2$  and  $L_\infty$  norms are used as performance measures in the model reduction and linearisation examples, the techniques could easily be extended to include the  $L_1$  norm or other performance measures if so desired.

### 7.3 EA Based Methodologies for System Identification and Modelling

$L_2$  norm based evolutionary black-box and clear-box system identification methods have been developed. Results obtained shown that the EA has offered a more accurate model than that identified from LMS-based technique. Evolutionary  $L_\infty$  norm black-box identification for robust control applications has also been developed. This method is applicable to both continuous and discrete-time systems and is capable of obtaining an optimal nominal model and an uncertainty bounding function.

For practical engineering systems, an evolution based grey-box modelling technique has been developed. Such a grey-box model utilises a physical law dominated global clear-box structure, with local black-boxes to include unmeasurable nonlinearities as coefficient models of the clear-box. This method has overcome the deficiency of operating point 'dependency' occurring in clear-box models of nonlinear systems. The evolution can start from empirical models, making best use of existing knowledge on a practical system. Experimental examples have shown that the proposed grey-box offers a more accurate model than that identified from a conventional LMS method. Evolution based closed-loop system identification has also been studied. The technique refines an open-loop identified clear-box model, within a relatively short processing time and shows good potential for on-line adaptation.

### 7.4 EA Based Methodology for LTI Control System Design Unification and Automation

Unification of linear control system designs by evolutionary computation has been studied in this thesis, for both the time and the frequency domains under performance satisfactions. It has been shown that the technique can accommodate many practical requirements imposed by engineering specifications, since it does not require differentiation of the performance index. The ULTIC strategy has been extended to control along a trajectory of operating points of nonlinear systems. Apart from following a fixed step reference, time-varying reference or model following control can also be realised by an ULTIC system.

Physical implementation of a coupled nonlinear liquid-level system has shown that a MIMO ULTIC system is easily realisable and offers good performance.

To enable a control engineer to interplay with design objectives, a multi-objective evolutionary algorithm has been incorporated in the ULTIC design, satisfying the ‘minimum-commitment’ principle at the CAD stage. A parallel EA has been implemented and a speedup of near-linear pipelinability and NP features have been observed from the parallel EA.

A direct convolution method for control system design from plant step response data, bypassing the system identification stage has been developed. Validation against linear and nonlinear plants has shown that the performance of the controller evolved from the response data of an inherently nonlinear plant is better than that designed from an identified 1<sup>st</sup> order model. This approach has offered a step towards autonomy in building control systems.

## **7.5 Future Perspectives**

### **Grey-Box Modelling with a Generic Black-Box Structure**

The evolutionary grey-box modelling technique developed in Chapter 4 has shown its ability in combating unstructured uncertainties usually found in practical systems. The uncertainties inherent in practical systems could be further combated when  $L_\infty$  norm or mixed time and frequency domain criteria are incorporated into this framework. Clearly, the black-box local learning can be extended to include neural network, nonlinear autoregressive moving average or other types of black-box model structures. Alternatively, genetic programming can also be incorporated to evolve the entire grey-box model using black-boxes and clear-boxes as symbolic building blocks.

## **Adaptive and Autonomous ULTIC Systems**

For adaptive control, the controller parameters are continuously adjusted to accommodate the changes in the plant parameters. There has been extensive research on adaptive control resulting from the need of auto-pilots for high performance aircraft since early 1950s (Åström and Wittenmark, 1989).

As illustrated in Figure 4.16, EAs can be used for on-line closed-loop system identification and hence the potential for on-line controller tuning. The ULTIC system developed in Chapter 5 can be evolved relatively rapidly and can thus help with adaptive control. Since the speed of variation of the plant parameters and the structure is usually much slower than the plant dynamics (Åström and Wittenmark, 1989, Ng, 1995), this identification and design phase can be allowed to proceed at a slower speed than that of the control signal provision. This becomes more achievable if micro hybrid EA and parallel EA studied in this thesis are adopted.

A further step towards autonomy is to obtain an ULTIC controller directly from off or on-line data, as this bypasses the system identification phase. Here, convolution operation can be extended to include variable types of input excitations. The long-term goal of this technique is to achieve the 'Plug and Play' design, i.e., to generate a digital ULTIC system on-line by plugging the response based design automator directly to the plant. The control system design task would then be tremendously eased, with added manufacturing productivity and quality and hence added competitiveness and wealth creation.

## REFERENCES

- Anderson, B. D. O. (1986). Weighted Hankel-norm approximation: Calculation of bounds, *System and Control Letters*, No. 7, 247-255.
- Anderson, J. H. (1967). Geometric approach to the reduction of dynamical systems, *IEE Proc. Control Theory Appl., Pt. D*, Vol. 114, No. 7, 1014-1018.
- Åström, K. J., and Bohlin, B. (1965). Numerical identification of linear dynamic systems from normal operating records, *IFAC Symposium on Self-Adaptive Systems*, Teddington, England; Also in Hammond, P. H., ed., *Theory of Self-Adaptive Control Systems*, Plenum Press, New York.
- Åström, K. J., and Wittenmark, B. (1989). *Adaptive Control*, Addison-Wesley, Reading, MA.
- Bacon, B. J., and Schmidt, D. K. (1988). Fundamental approach to equivalent system analysis, *J Guidance, Control and Dynamics*, Vol. 11, No. 6, 527-534.
- Baker, J. E. (1985). Adaptive selection methods for genetic algorithms, *Proceedings of an Int. Conf. on Genetic Algorithms and their Applications*, 101-111.
- Barker, H. A. (1995). Open environments and object-oriented methods for computer-aided control system design, *Control Eng. Practice*, Vol. 3, No. 3, 347-356.
- Beck, J. V., and Arnold, K. J. (1977). *Parameter Estimation in Engineering and Science*, John Wiley & Sons Ltd.
- Bellman, R. (1957). *Dynamic Programming*, Princeton University Press, Princeton, NJ.
- Caponetto, R., Fortuna, L., Muscato, G., and Xibilia, M. G. (1994). Genetic algorithms for controller order reduction, *Proc. 1st IEEE Conf. Evolutionary Computation*, IEEE World Congress on Computational Intelligence, Orlando, FL, Vol. 2, 724-729.
- Chiang, R. Y., and Safonov, M. G. (1992). *Robust Control Toolbox*, The MathWorks, Inc.
- Chipperfield, A. J., and Fleming, P. J. (1995). Gas turbine engine controller design using multiobjective genetic algorithms, *First Int. Conf. on GAs in Eng. Syst.: Innovations and Appl.*, Sheffield, UK, 214-219.
- Cluett, W. R., and Wang, L. (1991). Modelling and robust controller design using step response data, *Chemical Eng. Science*, Vol. 56, 2065-2077.

- Costello, D. J., and Gawthrop, P. J. (1995). Physical-model based control: experiments with a stirred-tank heater, *Technical report* CSC-95003 (<http://www.mech.gla.ac.uk/control/reports.html>), Centre for Systems and Control, University of Glasgow.
- Dakev, N. V., Whidborne, J. F., and Chipperfield, A. J. (1995).  $H_\infty$  design of an EMS control system for a maglev vehicle using evolutionary algorithms, *First Int. Conf. on GAs in Eng. Syst.: Innovations and Appl.*, Sheffield, UK, 226-231.
- Davies, W. D. T. (1970). *System Identification for Self-Adaptive Control*, John Wiley & Sons Ltd.
- Doyle, J. C., and Stein, G. (1981). Multivariable feedback design: Concepts for a classical/modern synthesis, *IEEE Trans. Automatic Control*, Vol. 26, No. 1, 4-16.
- Doyle, J. C., Francis, B., and Tannenbaum, A. (1992). *Feedback Control Theory*, Macmillan Publishing Company, New York.
- Fleming, P. J., and Fonseca, C. M. (1993). Genetic algorithms in control systems engineering, *Proc 12th IFAC Triennial World Congress*, Sydney, Australia, Vol. 2, 383-390.
- Flexible Intelligence Group, (1995). *User's manual of FlexTool(GA)*, L.L.C.
- Fogel, D. B. (1995). *Evolutionary Computation*, IEEE Press, Piscataway, NJ.
- Fonseca, C. M., and Fleming, P. J. (1993). Genetic algorithms for multiobjective optimisation: Formulation, Discussion and Generalisation, *Proc. of fifth Int. Conf. on Genetic Algorithms*, Morgan-Kaufman, 416-423.
- Fonseca, C. M., and Fleming, P. J. (1995). Multiobjective genetic algorithms made easy: Selection, sharing and mating restriction, *First Int. Conf. on GAs in Eng. Syst.: Innovations and Appl.*, Sheffield, UK, 45-52.
- Fonseca, C. M., Mendes, E. M., Fleming, P. J., and Billings, S. A. (1993). Non-linear model term selection with genetic algorithms, *In IEE/IEEE Workshop on Natural Algorithms in Signal Processing*, Essex, Vol. 2, 27/1-27/8.
- Franklin, G. F., Powell, J. D., and Emami-Naeini, A. (1991). *Feedback Control of Dynamic Systems*, Addison-Wesley.
- Gawthrop, P. J. (1995). Continuous-time local state local model network, *Proc. IEEE conf. on System, Man and Cybernetics*, Vancouver, British Columbia.

- Gawthrop, P. J., Nihtilä, M. T., and Rad, A. B. (1989). Recursive parameter estimation of continuous time systems with unknown time delay, *Control Theory and Advanced Technology*, Vol. 5, 227-248.
- Gawthrop, P. J., Jezek, J., Jones, R. W., and Sroka, I. (1993). Grey-box model identification, *Control-Theory and Advanced Tech.*, Vol. 9, No. 1, 139-157.
- Glover, K. (1984). All optimal Hankel-norm approximations of linear multivariable systems and their  $L_2$ -error bounds, *Int. Journal of Contr.*, Vol. 39, No. 6, 1115-1193.
- Goh, S. J., Gu, D. W., and Man, K. F. (1996). Multi-layer genetic algorithms in robust control system design, *Int. Conf. on Control'96, Special Session on Evolutionary Algorithms for Control Engineering*, University of Exeter, UK, 699-704.
- Goldberg, D. (1989). *Genetic algorithms in searching, optimisation and machine learning*, Reading, MA: Addison-Wesley.
- Goldberg, D., and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization, *GAs and their Applications: Proc. of the Second Int. Conf. on GAs*, 41-49.
- Gollee, H., Hunt, K. J., Donaldson, N., Jarvis, J. C. and Kwende, M. K. N. (1994). A mathematical analogue of electrically stimulated muscle using local model networks, *Proc. 33<sup>rd</sup> IEEE Conf. on Decision and Control*, Lake Buena Vista, Florida, 1879-1880.
- Golten, J. and Verwer, A. (1991). *Control System Design and Simulation*, Mcgraw-Hill Book Company, England.
- Gong, M., and Murray-Smith, D. J. (1993). Model reduction by an extended complex curve-fitting approach, *Trans. Inst. MC*, Vol. 15, No. 4, 188-198.
- Gordon, S., and Whitley, D. (1993). Serial and parallel genetic algorithms as function optimisers, *Proc. of the fifth Int. Conf. on Genetic Algorithms*, San Mateo, 177-183.
- Grace, A. (1992). *Optimisation Toolbox User's Guide*, The MathWorks, Inc.
- Grefenstette, J. J. (1986). Optimisation of control parameters for genetic algorithm, *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 16, No. 1, 122-128.
- Gu, G., and Khargonekar, P. P. (1992). Linear and nonlinear algorithms for identification in  $H_2$  with Error Bounds, *IEEE Trans. on Automatic Control*, Vol. 37, No. 7, 953-963.
- Guan, K. X., and MacCallum, K. J. (1996). Adopting a minimum commitment principle for computer aided geometric design systems, *In Artificial Intelligence in Design '96* (Gero, J. S. and Sudweeks, F., eds), Kluwer Academic Publishers, 623-639.

- Helmicki, A., Jacobson, C., and Nett, C. (1991). Control oriented system identification: A worst-case/deterministic approach in  $H_\infty$ , *IEEE Trans. on Automatic Control*, Vol. 36, No. 10, 1163-1176.
- Helmicki, A., Jacobson, C., and Nett, C. (1993). Least squares methods for  $H_\infty$  Control oriented system identification, *IEEE Trans. on Automatic Control*, Vol. 38, No. 5, 819-826.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor.
- Homaifar, A., and McCormick, E. (1992). Full design of fuzzy controllers using genetic algorithms, *Proc. SPIE Conf. on Neural and Stochastic Methods in Image and Signal Processing*, Vol. 1766, 393-404.
- Horn, J., Nafpliotis, N., and Goldberg, D. (1994). A niched pareto genetic algorithm for multiobjective optimisation, *Proc. First IEEE Conf. on Evolutionary Computation*, IEEE World Cong. on Computational Intelligence, Vol. 1, 82-87.
- Hunt, K. J. (1992). Polynomial LQG and  $H_\infty$  controller synthesis - A genetic algorithm solution, *Proc. 31st IEEE Conf. on Decision and Control*, Tucson, AZ, Vol. 4, No. 865, 3604-3609.
- Johansen, T. A. and Foss, B. A. (1992). A NARMAX model representation for adaptive control based on local models, *Modelling, Identification and Control*, Vol. 13, No. 1, 25-39.
- Johansen, T. A. and Foss, B. A. (1993). Constructing NARMAX models using ARMAX models, *Int. Journal of Contr.*, Vol. 58, 1125-1153.
- Karr, C. L. (1992). An adaptive system for process control using genetic algorithms, *In Int. Sym. on Artificial Intel. in Real-Time Contr.*, Delft, The Netherlands, 585-590.
- Kemna, A. H., and Mellichamp, D. A. (1995). Identification of combined physical and empirical-models using nonlinear *a-priori* knowledge, *Contr. Eng. Pract.*, Vol. 3, No. 3, 375-382.
- Kosut, R. L., Lau, M. K., Boyd, S. P. (1992). Set-membership identification of systems with parametric and nonparametric uncertainty, *IEEE Trans. on Automatic Control*, Vol. 37, No. 7, 929-941.



- Krishnakumar K., and Satyadas, A. (1996). Evolving multiple fuzzy models and its application to an aircraft control problem, *Genetic Algorithms in Engineering and Computer Science*, Winter *et al.*, (Eds), John Wiley & Sons Ltd.
- Kristinsson, K., and Dumont, G. A. (1992). System identification and control using genetic algorithms, *IEEE Trans. Syst., Man and Cyber.*, Vol. 22, No. 5, 1033-1046.
- Kwong, S., Ng, A. C. L., and Man K. F. (1995). Improving local search in genetic algorithms for numerical global optimization using modified GRID-point search technique, *First Int. Conf. on Gas in Eng. Syst.: Innovations and Appl.*, Sheffield, UK, 419-423.
- Latham, G. A., and Anderson, B. D. O. (1986). Frequency weighted optimal hankel-norm approximation of stable transfer function, *System and Control Letters*, No. 5, 229-236.
- Lawrence, D. A., and Rugh, W. J. (1994). Input-output pseudo-linearization for nonlinear system, *IEEE Trans. Autom. Control*, Vol. 39, No. 11, 2207-2218.
- Levy, E. C. (1959). Complex-curve fitting, *IRE Trans. on Automatic Control*, AC-4, 37-44.
- Li, Y. (1995). Modern information technology for control systems design and implementation, *Proc. 2nd Asia-Pacific Conference on Control and Measurement*, Chongqing, China, 17-22.
- Li, Y. (1996). *Neural and Evolutionary Computing IV*, Lecture Notes (2RPX), Dept. of Electronics and Electrical Engineering, University of Glasgow.
- Li, Y., and Häußler, A. (1996). Artificial evolution of neural networks and its application to feedback control, *Artificial Intelligence in Engineering*, Vol. 10, No. 2, 143-152.
- Li, Y., Ng, K. C., Tan, K. C., Gray, G. J., McGookin, E. W., Murray-Smith, D. J., and Sharman, K. C. (1995a). Automation of linear and nonlinear control systems design by evolutionary computation, *Proc. IFAC Youth Automation Conf.*, Beijing, China, 53-58.
- Li, Y., Tan, K. C., Ng, K. C., and Murray-Smith, D. J. (1995b). Performance based linear control system design by genetic evolution with simulated annealing, *34th IEEE Conf. Decision and Control*, New Orleans, LA, 731-736.
- Li, Y., Ng, K. C., Murray-Smith, D. J., Sharman, K. C., and Gray, G. J. (1996a). Genetic algorithm automated approach to design of sliding mode control systems, *Int. Journal of Contr.*, Vol. 63, No. 4, 721-739.
- Li, Y., Tan, K. C., and Marionneau, C. (1996b). Direct design of linear control systems from plant I/O data using parallel evolutionary algorithms, *Int. Conf. on Control'96*,

- Special Session on Evolutionary Algorithms for Control Engineering*, University of Exeter, UK, 680-686.
- Li, Y., Tan, K. C., and Gong, M. R. (1997). Global structure evolution and local parameter learning for control system model reductions, *Evolutionary Algorithms in Engineering Applications*, D. Dasgupta and Z. Michalewicz (Eds.), Springer Verlag.
- Linkens, D. A., and Abbod, M. F. (1992). A real-time genetic algorithm for fuzzy control, *IEEE Collg. On GAs for Contr. Sys. Eng.*, 9/1-9/4.
- Ljung, L. (1987). *System Identification - Theory for the User*, Prentice-Hall, Englewood Cliffs, N.J.
- Ljung, L. (1992). *System Identification Toolbox*, The Math Works, Inc.
- Ljung, L., and Söderström, T. (1983). *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, Mass.
- Maclay, D., and Dorey, R. (1993). Applying genetic search techniques to drivetrain modeling, *IEEE Contr. Syst. Mag.*, Vol. 13, No. 3, 50-55.
- MathTools, (1997). *Matcom V2.01- A Matlab to C++ Compiler*, The MathTools, Inc.
- MathWorks, (1992). *Matlab Reference Guide*, The MathWorks, Inc.
- Michalewicz, Z. (1994). *Genetic Algorithms + Data Structure = Evolutionary Programs*, Springer-Verlag, Berlin, 2<sup>nd</sup> Edition.
- Murdock, T. M., Schmitendorf, W. E., and Forrest, S. (1991). Use of a genetic algorithm to analyze robust stability problems, *Proc. of the 1991 American Control Conference*, Massachussets, Vol. 1, 886-889.
- Murray-Smith, R. (1994). *A Local Model Network Approach to Nonlinear Modelling*, Ph.D. Thesis, University of Strathclyde, Glasgow, Scotland.
- Ng, K. C. (1995). *Switching Control Systems and Their Design Automation via Genetic Algorithms*, Ph.D. Thesis, Dept. of Electronics and Electrical Engineering, University of Glasgow, Scotland.
- Osyczka, A. (1984). *Multicriterion Optimisation in Engineering*, Ellis Horwood, Chichester.
- Patton, R. J., and Liu, G. P. (1994). Robust control design via eigenstructure assignment, genetic algorithms and gradient-based optimisation, *IEE Proc. Control Theory Appl.*, Pt. D, 141, 202-208.
- Patton, R. J., Chen, J., and Liu, G. P. (1995). Robust fault detection of dynamic systems via genetic algorithms, *First Int. Conf. on GAs in Eng. Syst.: Innovations and Appl.*, Sheffield, UK, 511-516.

- Pintelon, R., Guillaume, P., Rolain, Y., Schoukens, J., and Van Hamme, H. (1994). Parametric identification of transfer functions in the frequency domain - A survey, *IEEE Trans. Auto. Control*, Vol. 39, No. 11, 2245-2259.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1994). *Numerical Recipes in C*, Cambridge University Press, Cambridge.
- Renders, J. M., and Bersini, H. (1994). Hybridizing genetic algorithms with hill-climbing methods for global optimisation: two possible ways, *Proc. 1st IEEE Int. Conf. Evolutionary Computing*, IEEE World Cong. Comp. Intel., Orlando, Vol. 1, 312-317.
- Sedgewick, R. (1988). *Algorithms*, Addison-Wesley, Reading, MA., 2<sup>nd</sup> Edition.
- Sharman, K. C., and Esparcia-Alcázar, A. I. (1993). Genetic evolution of symbolic signal models, *Proc. IEE/IEEE Workshop on Natural Algorithms in Signal Processing*, University of Essex.
- Sharman, K. C., Esparcia-Alcázar, A.I., and Li, Y. (1995). Evolving signal processing algorithms by genetic programming, *First Int. Conf. on GAs in Eng. Syst.: Innovations and Appl.*, Sheffield, UK, 473-480.
- Sirag, D., and Weisser, P. (1987). Toward a unified thermo-dynamic genetic operator, *Proc. of the 2<sup>nd</sup> international Conf. on Genetic Algorithms*, 116-122.
- Söderström, T., and Stoica, T. (1989). *System Identification*, Prentice-Hall International, London.
- Spears, W., and DeJong, K. (1991). An analysis of multi-point crossover, *Foundations of Genetic Algorithms*, San Mateo, 301-315.
- Srinivas, M., and Patnaik, L. M. (1994). Genetic algorithms: A survey, *IEEE Computer*, Vol. 27, No. 6, 17-26.
- Sugie, T., and Tanai, Y. (1995).  $L_{\infty}$  identification in the frequency domain based on the set-membership approach, *Proc. 34<sup>th</sup> IEEE Conf. Decision and Control*, New Orleans, LA, Vol. 3, 3016-3021.
- Tan, K. C., and Li, Y. (1996).  $L_{\infty}$  identification and model reduction using a learning genetic algorithm, *Int. Conf. on Control'96*, University of Exeter, UK, 1125-1130.
- Tan, K. C., and Li, Y. (1997b). Multi-objective genetic algorithm based time and frequency domain design unification of control systems, *IFAC Int. Sym. on Artificial Intelligence in Real-Time Control*, Kuala Lumpur, Malaysia. (submitted)
- Tan, K. C., Gong, M. R., and Li, Y. (1996). Evolutionary linearisation in the frequency domain, *Electronics letters*, Vol. 32, No. 1, 74-76.

- Tan, K. C., Li, Y., Murray-Smith, D. J., and Sharman, K. C. (1995). System identification and linearisation using genetic algorithms with simulated annealing, *First Int. Conf. on GAs in Eng. Syst.: Innovations and Appl.*, Sheffield, UK, 164-169.
- Tan, K. C., Li, Y., Gawthrop, P. J., and Glidle, A. (1997). Evolutionary grey-box modelling for practical systems, *Second IEE Int. Conf. on GA in Eng. Syst. Innovations and Appl.*, Glasgow, UK. (submitted)
- Thomson, D. G., and Bradley, R. (1988). Validation of helicopter mathematical models by comparison of data from Nap-of-the-Earth flight tests and inverse simulation, *Proc. 14<sup>th</sup> European Rotorcraft Forum*, Milan, Paper No. 87.
- Vandemolengraft, M. J. G., Veldpaus, F. E., and Kok, J. J. (1994). An optimal estimation method for nonlinear mechanical systems, *J. Dyn. Syst. Meas. Contr., Trans. ASME*, Vol. 116, No. 1, 805-810.
- Wang, P., and Kwok, D. P. (1992). Auto-tuning of classical PID controllers using an advanced genetic algorithm, *Proc. Int. Conf. Ind. Elect., Contr., Instrum. and Automation*, Vol. 3, No. 286, 1224-1229.
- Warwick, K. (1984). A new approach to reduced-order modelling, *Proc. IEE Control Theory and Applications, Pt. D*, Vol. 131, No. 2, 74-78.
- Xue, D., and Atherton D. P. (1993). An optimal reduction algorithm for discrete-time systems, *Proc. IFAC 12th Triennial World Congress*, Sydney, Australia, 821-824.
- Yang, Z. J., Hachino, T., and Tsuji, T. (1996). Model reduction with time delay combining the least-squares method with the genetic algorithm, *IEE Proc. Cont. Theory Appl.*, Vol. 143, No. 3. 247-254
- Yang, Z. J., Hachino, T., and Tsuji, T. (1997). On-line identification of continuous time-delay systems combining least-squares techniques with a genetic algorithm, *Int. Journal of Contr.*, Vol. 66, No. 1, 23-42.
- Yoa, L., and Sethares, W. A. (1994). Nonlinear parameter estimation via the genetic algorithm, *IEEE Trans. Sig. Proc.*, Vol. 42, No. 4, 927-935.
- Zames, Z. (1966). On the input-output stability of time-varying nonlinear feedback systems, Parts I and II, *IEEE Trans. Auto. Contr.* AC-11, No. 2 & 3, 228-238 & 465-476.
- Zhang, L. J. (1995). *A Study on Fuzzy Neural Network Technology & its Applications in Traffic Control of ATM Networks*, Ph.D. Thesis, University of Qin Hua, China.

## References

- Zhou, K. (1995). Frequency-weighted  $L_2$  norm and optimal Hankel norm model reduction, *IEEE Trans. Automatic Control*, Vol. 40, No. 10, 1687-1699.
- Zhou, T., and Kimura, H. (1993). Time domain identification for robust control, *System and Control letters*, Vol. 20, No. 3, 167-173.
- Zhou, T., and Kimura, H. (1994). Simultaneous identification of nominal model, parametric uncertainty and unstructured uncertainty for robust control, *Automatica*, Vol. 30, No. 3, 391-402.

Appendix A

THE COUPLE LIQUID-TANK NONLINEAR MODEL

The coupled liquid-tank system that has been experimented in this thesis is shown in the following diagram. Here the tanks are linked through a coupling pipe of an equivalent orifice area  $a_1$ ; the equivalent discharging area of Tank 2 is modelled by  $a_2$ ; The liquid levels in Tank 1 and Tank 2 are  $h_1$  and  $h_2$ , respectively;  $H_3$  is the equivalent height of both the coupling and discharging pipes.

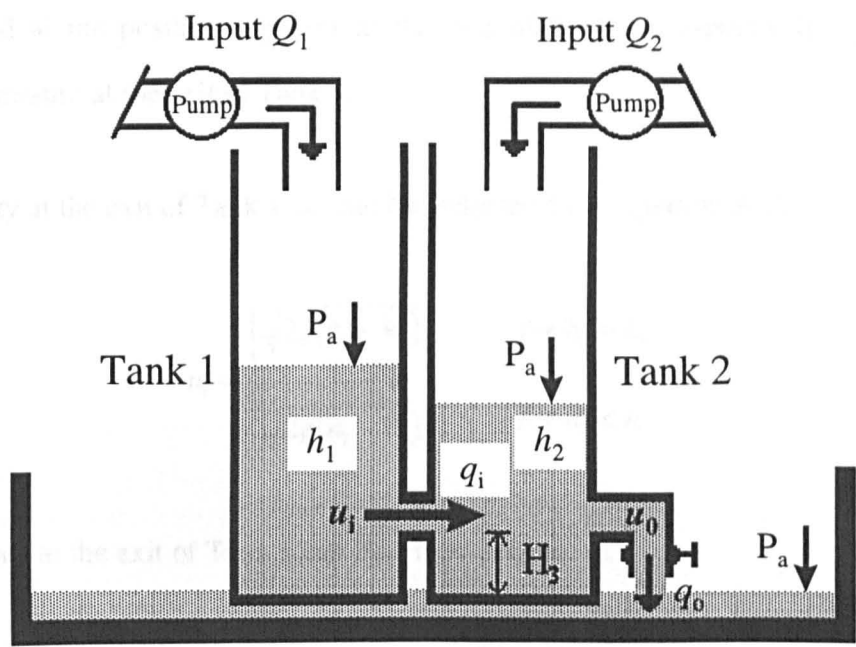


Figure A1 A coupled liquid-tank system

**Bernoulli's Theorem:** In the steady motion of liquid, the quantity

$$P + \frac{1}{2} \rho v^2 + \rho gh \tag{A-1}$$

has the same value at every point of the same streamline, where  $P$ ,  $\rho$  and  $v$  are the pressure, density and speed, respectively;  $h$  is the height and  $g$  the gravitational constant. Suppose

that the volumes of the two tanks are large enough and the flow rates at their exits,  $q_i$  and  $q_o$  are small, such that the flow inside the tank is negligibly small. By assuming both tanks have the same cross-sectional area and applying the Bernoulli's theory to Tank 1, we can have

$$P_a + \rho g(h_1 - H_3) = \frac{1}{2} \rho u_i^2 + \rho g(h_2 - H_3) + P_a \quad (\text{A-2})$$

where,  $P_a$  is the ambient air pressure acting on the liquid surface;

$u_i$  is the flow velocity at the exit of Tank 1

Here  $\rho g(h_1 - H_3)$  and  $\rho g(h_2 - H_3)$  are the pressures exerted by water at the height of  $H_3$  in Tank 1 and at the position adjacent to the exit of Tank 1, respectively;  $\frac{1}{2} \rho u_i^2$  is the dynamic pressure at the exit of Tank 1.

The velocity at the exit of Tank 1,  $u_i$ , can be deducted from equation A-2,

$$u_i = \begin{cases} \sqrt{2g(h_1 - h_2)} & \text{for } h_1 > h_2 \\ \sqrt{2g(h_2 - h_1)} & \text{for } h_1 < h_2 \end{cases} \quad (\text{A-3})$$

The flow rate at the exit of Tank 1 can thus be expressed as

$$q_i = C_1 a_1 u_i = C_1 a_1 \sqrt{2g|h_1 - h_2|} \quad (\text{A-4})$$

where  $a_1$  is the orifice cross sectional area of Tank 1 and  $C_1$  the equivalent discharge constant of Tank 1.

Similarly, applying Bernoulli's theory to the Tank 2, we have

$$P_a + \rho g(h_2 - H_3) = \frac{1}{2} \rho u_o^2 + P_a \quad (\text{A-5})$$

The velocity at the exit of Tank 2 can be derived from equation A-5,

$$u_o = \sqrt{2g(h_2 - H_3)} \quad (\text{A-6})$$

and the flow rate at exit of Tank 2 is

$$q_o = C_2 a_2 u_o = C_2 a_2 \sqrt{2g|h_2 - H_3|} \quad (\text{A-7})$$

where  $a_2$  is the orifice cross sectional area of Tank 2 and  $C_2$  the equivalent discharge constant of Tank 2.

The dynamic equations of the system can thus be derived by taking the flow balances for each of the tank. The rate of change of fluid volume can be determined from the difference between input and output flows for Tank 1 and Tank 2 respectively, i.e.,

$$\frac{dV_1}{dt} = A \frac{dh_1}{dt} = Q_1 v_1 - q_i \quad (\text{A-8})$$

$$\frac{dV_2}{dt} = A \frac{dh_2}{dt} = Q_2 v_2 + q_i - q_o$$

where  $V_1$  and  $V_2$  are the volumes of liquid in Tank 1 and Tank 2, respectively;  $Q_1$  and  $Q_2$  the input flow rate per actuating volt of the power amplifiers for Tank 1 and Tank 2, respectively.

From equations A-4, A-7 and A-8, the nonlinear coupled liquid-tank model is obtained,

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} -\text{sgn}(h_1 - h_2) \frac{C_1 a_1}{A} \sqrt{2g|h_1 - h_2|} \\ \text{sgn}(h_1 - h_2) \frac{C_1 a_1}{A} \sqrt{2g|h_1 - h_2|} - \frac{C_2 a_2}{A} \sqrt{2g|h_2 - H_3|} \end{bmatrix} + \begin{bmatrix} \frac{Q_1}{A} & 0 \\ 0 & \frac{Q_2}{A} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (\text{A-9})$$



## Appendix B

## THE COUPLE LIQUID-TANK LINEAR MODEL

A linear model representation of the coupled liquid-tank system in Figure A1 could be derived by linearising around an equilibrium point of the nonlinear model of Equation A-9 obtained in Appendix A. This is achieved by considering only small variations  $q_1$  in  $Q_1$ ,  $q_2$  in  $Q_2$ ,  $q_{ii}$  in  $q_i$ ,  $q_{oo}$  in  $q_o$ ,  $h_{11}$  in  $h_1$  and  $h_{22}$  in  $h_2$ .

In the steady-state,  $Q_1 v_1 = q_i$ ,  $Q_2 v_2 + q_i = q_o$  and

$$A \frac{dh_{11}}{dt} = q_1 v_1 - q_{ii} \quad (\text{B-1})$$

$$A \frac{dh_{22}}{dt} = q_2 v_2 + q_{ii} - q_{oo} \quad (\text{B-2})$$

By taking the first derivative upon Equations (A-4) and (A-7) in Appendix A respectively,  $q_{ii}$  and  $q_{oo}$  can be obtained,

$$q_{ii} = \frac{1}{2} \sqrt{2g} C_1 a_1 \left( \frac{h_{11} - h_{22}}{\sqrt{h_1 - h_2}} \right) \quad (\text{B-3})$$

$$q_{oo} = \frac{1}{2} \sqrt{2g} C_2 a_2 \left( \frac{h_{22}}{\sqrt{h_2 - H_3}} \right) \quad (\text{B-4})$$

where  $a_1$ ,  $a_2$ ,  $C_1$ ,  $C_2$ ,  $H_3$  and  $g$  are defined as in Appendix A. Substituting Equations (B-1) and (B-2) into (B-3) and (B-4) respectively,

$$A \frac{dh_{11}}{dt} = q_1 v_1 - \frac{1}{2} \sqrt{2g} C_1 a_1 \left( \frac{h_{11} - h_{22}}{\sqrt{h_1 - h_2}} \right) \quad (\text{B-5})$$

$$A \frac{dh_{22}}{dt} = q_2 v_2 + \frac{1}{2} \sqrt{2g} C_1 a_1 \left( \frac{h_{11} - h_{22}}{\sqrt{h_1 - h_2}} \right) - \frac{1}{2} \sqrt{2g} C_2 a_2 \left( \frac{h_{22}}{\sqrt{h_2 - H_3}} \right) \quad (\text{B-6})$$

By rearranging Equations (B-5) and (B-6), the following state-space linear model of the coupled tanks system is obtained

$$\begin{bmatrix} \dot{h}_{11} \\ \dot{h}_{22} \end{bmatrix} = \begin{bmatrix} -\frac{k_1}{A} & \frac{k_1}{A} \\ \frac{k_1}{A} & -\frac{k_1 + k_2}{A} \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{22} \end{bmatrix} + \begin{bmatrix} \frac{q_1}{A} & 0 \\ 0 & \frac{q_2}{A} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (\text{B-7})$$

where  $k_1 = \frac{\sqrt{2g} C_1 a_1}{2\sqrt{h_1 - h_2}}$

$$k_2 = \frac{\sqrt{2g} C_2 a_2}{2\sqrt{h_2 - H_3}}$$

